

**3rd International Conference on
Computer and IT Applications in the Maritime Industries**

COMPIT'04

Siguënza, 9-12 May 2004

Volker Bertram, Manuel Armada (Ed.)



Sponsored by



This work relates to Department of the Navy Grant N00014-04-1-1055 issued by the Office of Naval Research Global. The United States Government has a royalty-free license throughout the world in all copyrightable material contained herein.

Index

Volker Bertram <i>Towards Intelligent Simulation-based Ship Design</i>	5
Marco Ferrando, Andrea Lommi, Antonio Traverso <i>Use of Genetic Algorithms in Propeller Foil Design</i>	17
Jörn Hinnenthal ¹ , Stefan Harries <i>A Systematic Study on Posing and Solving the Problem of Pareto Optimal Ship Routing</i>	27
Otto Millbourn <i>Concurrent Engineering through the Supply Chain from Ship Owner to Supplier</i>	36
Hans Wuttke, Andreas Schwill, Frank Domeyer <i>An E-Commerce Platform for Engineering Services in Shipbuilding</i>	42
Tobias Haack, Stefan Krüger <i>Propulsion Plant Models for Nautical Manoeuvre Simulations</i>	52
Lawrence Henesey ² , Paul Davidsson, Jan A. Persson <i>Using Simulation in Evaluating Berth Allocation at a Container Terminal</i>	61
Eberhard Blümel, Leonid Novitski <i>Introduction to BALTPORTS-IT: Applications of Simulation and IT-Solutions in the Baltic Port Areas</i>	73
Peter Weiss ¹ , Yann Le Page, Frédéric Schom, Alain Fidani <i>Robot Applications in the Field of Shipbuilding</i>	81
Vedeesh Sahajpal <i>An intelligent GA based AIS FATDMA scheduler</i>	86
Jonathan M. Ross <i>A Practical Approach for Ship Construction Cost Estimating</i>	98
Robert Bronsart, Steffen Gau, Diane Luckau, Wolfgang Sucharowski <i>Communication in ship design networks</i>	111
Torsten Fischer, Hermann Gehring <i>A Multi-Agent based Approach for Solving the Vehicle Transshipment Problem</i>	122
Meelis Mäesalu, Jerzy Matusiak <i>Visualisation of Ship Motions in Waves and during Grounding for Simulators</i>	131
Peter Weiss ¹ , Fivos Andritsos, Frédéric Schom, Alain Fidani <i>Innovative Robotic Solutions for the Survey and Certification of Ships and Mobile Offshore Units</i>	140
Yuichi Sasaki, Hiroyuki Yamato, Shoichi Enomoto <i>Research on Industrial Engineering System by using Wearable PC</i>	148

¹ Partially supported by SENER

² Partially sponsored by Tribon Solutions

Patrick Hupe, Uwe Langbecker, Mubashir Aziz, Joachim W. Schmidt <i>Mobile and Web-based Services for Ship Operation and Survey: A Portal-centric Approach</i>	154
Jan Henrik Weychardt, Berend Bohlmann <i>FEM Supported Alignment of Power Train Components</i>	165
Evangelos K. Boulougouris, Apostolos D. Papanikolaou <i>Optimisation of the Survivability of Naval Ships by Genetic Algorithms</i>	175
Jan Jaap Nieuwenhuis, Ubald Nienhuis <i>Knowledge and Data Reuse in Ship System Design and Engineering</i>	190
Zbigniew Pietrzykowski, Janusz Uriasz <i>The Ship Domain in a Deep-Sea Area</i>	204
Bastiaan Veelo ² <i>Shape Modification of Hull Models in H-rep</i>	212
Sipke Vijver ¹ , Hotze Boonstra, Herbert J. Koelman, Erwin D. Stinstra <i>Applying Meta-Models to the Probabilistic Damage Stability</i>	223
Hugo Grimmelius, Douwe Stapersma, Paul Schulten <i>Ship Propulsion Control: A Fresh View</i>	236
Desmond Simpson, Ludwig Furstenberg <i>Distributing Operational Simulation Systems across South African Ports</i>	252
Jenny Coenen, Martin van Hees, Ubald Nienhuis <i>Knowledge-based Concurrent Engineering in One-off Ship Design</i>	263
Jean-Yves Pradillon <i>ODIGO - Modelling and Simulating Crowd Movement onboard Ships</i>	278
Uwe Langbecker, Ricardo Pereira <i>UTHLANDE – An Extensible Platform for Hydromechanic Calculation Modules</i>	290
Patrick Queutey, Michel Visonneau <i>Three-dimensional CFD Simulations Using a Free-Surface Capturing Strategy</i>	300
Manuel Armada, Pablo González de Santos, Manuel Prieto, Elena García, Teodor Akinfiev Roemi Fernández, Hector Montes, Samir Nabulsi, Luis Pedraza, Roberto Ponticelli, Javier Sarriá, Joaquín Estremera <i>State of the art in climbing and walking robots</i>	315
Antonio Pinto, Daniele Peri, Emilio F. Campana <i>Global Optimization Algorithms in Naval Hydrodynamics</i>	322
Virgil Mutu, Ovidiu Ionas <i>Computer Applications and Technologies at Ship Design Group Galati</i>	334
Bekir S. Türkmen ² , Osman Turan <i>An Application Study of Multi-Agent Systems in Multi-Criteria Ship Design Optimisation</i>	340

Rafael de Góngora <i>A New Collaborative Engineering Management Tool for Shipbuilding</i>	354
Martijn Neef, Anthonie van Lieburg <i>Applications of Self-Organizing Systems in Maritime Environments</i>	363
Christian Massow, Ivan Siksne-Pedersen <i>Computer Integrated Planning and Resource Management in Shipbuilding</i>	378
Patrick Couser, Andrew Mason, Garth Mason, Cameron R. Smith, Brian R. von Konsky <i>Artificial Neural Networks for Hull Resistance Prediction</i>	391
Reidar Tronstad, Antonio Rodriguez <i>Automation Tools in the Design Process</i>	403
Andreas Bortfeldt <i>A Heuristic for the Container Pre-Marshalling Problem</i>	419
David E. Hess, William E. Faller, Edward S. Ammeen, Thomas C. Fu <i>Neural Networks for Naval Applications</i>	430
Mario Dogliani, Dracos Vassalos, Tom Strang <i>Evacuation Notation – a New Concept to Boost Passenger Evacuation Performance in the Cruise Industry</i>	447
M.S. Seif, E. Jahanbakhsh <i>Neural Networks Model for Ship Maneuver</i>	459
Wilfried Abels, Stefan Krüger <i>Correlation Strategy for Propeller Excited Pressure Fluctuations</i>	470
Index of authors	479

Call for Papers COMPIT'05 at last page

Email the authors to obtain an electronic copy often containing figures in colour.

Towards Intelligent Simulation-Based Design

Volker Bertram, ENSIETA, Brest/France, volker.bertram@ensieta.fr

Abstract

Proposed models for the ship design process fail to convince. The classical design spiral model is frequently rejected by modern ship designers, but design is iterative and several basic elements are repeated used, albeit not always in same order. The paper gives some historical perspective before reviewing some modern developments in computer-aided ship design, drawing frequently, but not exclusively, on the experience of the author. Techniques covered include knowledge and decision, generation of solutions, analysis and visualization.

1. Introduction

The design process is often described as a design spiral following the initial model of *Evans (1959)*. This model has been "refined" by proposing spiraling cones, *Andrews (1981)*, or interconnected spirals, *Mistree et al. (1990)*, but in essence it was the spiral in disguise, Fig.1. This model for ship design has been rejected frequently, as the sequence of analyses and decisions is not as sequential as the spiral models suggest. Ship design seems to be instead a rather chaotic process of combining various analyses with 'intelligent' or 'creative' generation of solutions, sometimes in parallel, which are then analyzed, evaluated, possibly rejected or modified. The sequence of these elements in the design process changes even for absolutely the same design task from designer to designer. There is no absolute criterion what is the best design process sequence. While different designers may well apply the same tools (or at least tools of comparable performance power), some designers manage to get better results.

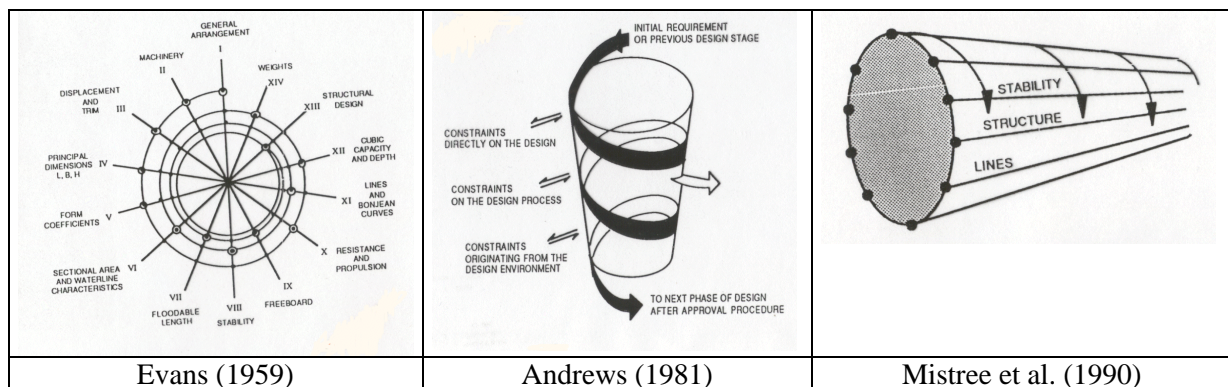


Fig.1: Ship design has been frequently described as a design spiral process

I see this in rough analogy to nature. The DNA is composed of the 4 same basic elements, whether it is the DNA of a bacterium or the DNA of a Nobel prize laureate. Apparently the proper combination of the same 4 simple elements can make a big difference. Similarly, one could use music as an analogue with a (relatively) small number of string, keys or notes used by masters and beginners alike, but with big differences. Rather than speculating whether computers in the (distant) future may outperform humans in the design process, I focus on the "basic elements" of ship design and highlight some recent developments. The ship design process remains computer-aided, but the computer accelerates the process and improves the "classical toolkit" of the designer. The following is neither mutually exclusive, nor collectively exhaustive. It is merely intended to bring some structure into the review. Basic elements in ship design are then for me:

1. Generation (creative, intelligent or systematic creation of solutions)
2. Analysis (evaluation, simulation, computation)
3. Visualization (representation, communication)
4. Decision (knowledge, identification, learning)

2. Generation

The generation of solutions is rather a creative task. While “Artificial Intelligence” is by now widely accepted as an additional useful tool also for naval architects, nobody contemplates (seriously) “Artificial Creativity”. And yet the computer may help us to generate better solutions, namely in the context of concept exploration and/or optimization, *Bertram (2003a)*.

A concept exploration models (CEM) generates a large set of candidate solutions by varying design variables. Each of these solutions is evaluated, stored and graphically displayed, so that the designer gets a feeling how certain variables influence the performance of the design, e.g. *Erikstad (1996)*. This preliminary “exploration” is often necessary to determine suitable optimization objectives and constraints. This approach, albeit under the misleading label of “multi-objective optimization” is realized in the modeFRONTIER optimization environment, e.g. *Abt et al. (2003)*. The approach is in reality a preliminary concept exploration generating many solutions and defining Pareto surfaces, Fig.2, which then allow the user to reject certain solutions (realizing at this stage constraints which were so far not explicitly stated) and finding an “optimum” solution (within the more or less simplified optimization model using more or less erroneous approximations for evaluating candidate properties). Despite remaining short-comings, optimization shells as modeFRONTIER or the German development DELPHI, e.g. *Bertram et al. (1998)*, *Gudenschwager (2003)*, allow the designer to focus on the design task and allow by now models of suitable complexity to indeed aid the practical design process. Setting up a suitable optimization model, rather than programming yet another optimization algorithm, is the task for the naval architect.

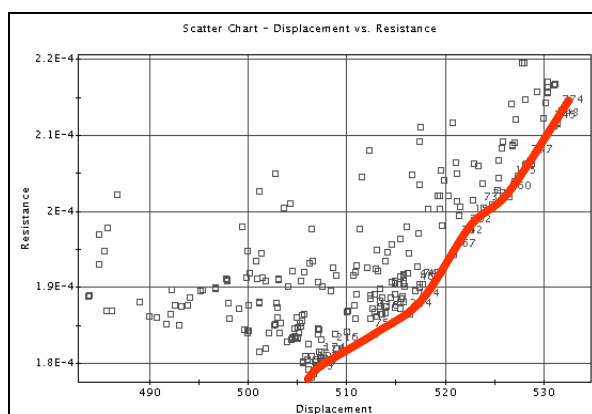


Fig.2: Result of form optimization with Pareto Frontier for FantaRoRo, source: TU Berlin

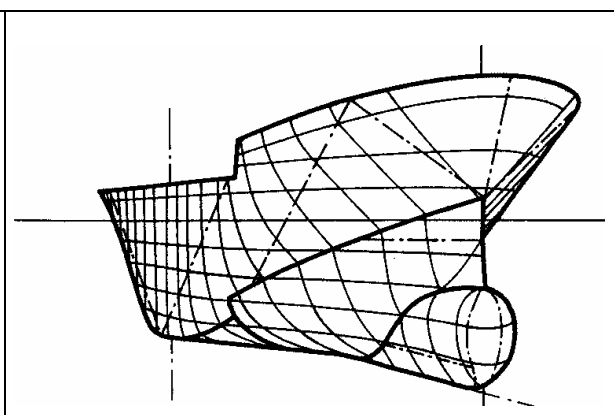


Fig.3: Schenzle's 'design for production' hull, *Bertram and El Moctar (2003)*

New and improved simulation tools can be integrated in optimization models allowing new or more realistic applications. Possibilities are as numerous as the rapidly developing simulation tools. Examples from my own research interest and experience include:

- Hull production costs are so far rarely quantified in the design. Bottom-up approaches (decomposing the production process into appropriate subtasks like profile and plate bending, welding in various positions and techniques, etc.) are needed to suitably reflect the influence of local hull shape changes, *Bertram and El Moctar (2003)*. *Rigo (2003)* presents how bottom-up approaches can now be applied successfully to support shipyard structural design work. Hulls optimized for production costs without hydrodynamic sacrifices appear feasible, Fig.3.
- Optimization uses often an economic criterion. Ships have been optimized in the past for building costs (“shipyard’s view”) or yield including operational cost and income (“ship owner’s view”). Environmental aspects have been largely neglected in optimization with the notable exception of wash, e.g. *Koushan et al. (2002)*. However, ship emissions cause indi-

rectly damage in terms of greenhouse effect, acid rain, eutrophication etc. The associated costs are ‘external costs’ which have to be considered on a scale of national or global economies. Legislation generally has the trend to internalize external costs. *Isensee and Bertram (2004)* present first steps in quantifying ship emissions in terms of external costs, thus opening the door for optimization studies including external costs (“state’s view”).

- Ship hull optimization benefits from advances in hull description, namely parametric hull description, and progress in CFD, e.g. *Harries (1998)*. A practical problem is that the hull description requires consideration of the intended optimization at the time of setting up the parametric model. Thus existing CAD descriptions are usually of little value and a new hull description (of an existing hull) is required for an optimization. A diploma thesis at ENSIETA compared three commercial CAD systems (NAPA, CATIA, Friendship) offering parametric descriptions using the a generic ro-ro ship (FantaRoRo) as test case, Fig.4, *Butruille (2002)*. Table I gives the (subjective) evaluation of *Butruille (2002)* who pointed out remaining need for research and development in parametric ship hull description. Also CFD methods employed feature still large (and variable) errors in predicting e.g. resistance values requiring care and often iterative refinements of the optimization model to obtain plausible results, e.g. *Abt et al. (2003)*. Progress in CFD method and experience in how to set up models should however improve the picture and let formal hull optimization at least for bulbous bows become a practical reality within the next decade.

Table I: Abbreviated table of evaluation for parametric ship hull description benchmark study, *Butruille (2002)*

	NAPA	FRIENDSHIP	CATIA
Ability to approximate given surface with parametric description	Not tested. Handling of parameters does not seem simple.	average (good for conventional forms)	good
Facility of parameterization (time needed for user)	quite slow	fast (GUI needed)	average (conventional forms)
Grid generation for CFD code	simple	simple	average
IGES export	good	good	average
Surface quality	average (too sensitive on changes in parameters)	average (some problems at junctions)	average (sometimes bad w/o apparent reason)
Robustness on parameter variations	poor	quite robust	poor
Proper program end in case of error in attempt to create parametric surface	no (sometimes program not closing in case of error)	yes (often without error message)	yes
Duration for surface generation	1 minute	some seconds	2-4 minutes

Ship routing (i.e. optimization of a ship’s course and speed) is a special application of optimization which couples naval architecture and ship operation. Both formal optimization and techniques of Artificial Intelligence (AI) have been proposed for ship routing. We distinguish between strategical and tactical routing. Strategical routing has typical time periods of hours or days, tactical routing has typical time periods of minutes. Tactical routing concerns e.g. collision avoidance, e.g. *Kasai and Bertram (1996)*, *Bertram (2000b)*. PrISM is a tactical ship routing system developed as a decision support system for the French navy in finding best course and speed for a frigate during helicopter landing, *Capitant and Bertram (2003)*. The system combines various AI techniques (genetic algorithms for seaway identification, fuzzy logic in evaluating safety level, expert system for criteria of safety) with a graphical user interface (GUI), Fig.5. By late 2003, the system was installed and subject to evaluation by the French navy. Spin-off modifications of the system to tugs and towed ships are subject to pending research applications. Progress in telecommunication, optimization

environments and practical AI tools motivate renewed research efforts for strategic routing, which after at least three decades of research can be considered as a “classical” optimization topic.

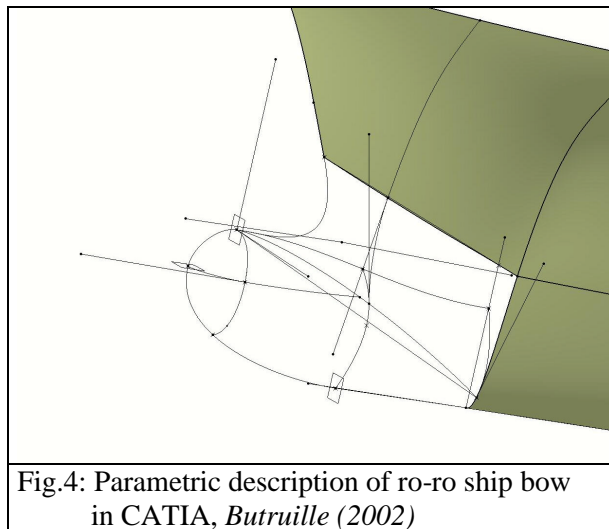


Fig.4: Parametric description of ro-ro ship bow in CATIA, *Butruille (2002)*

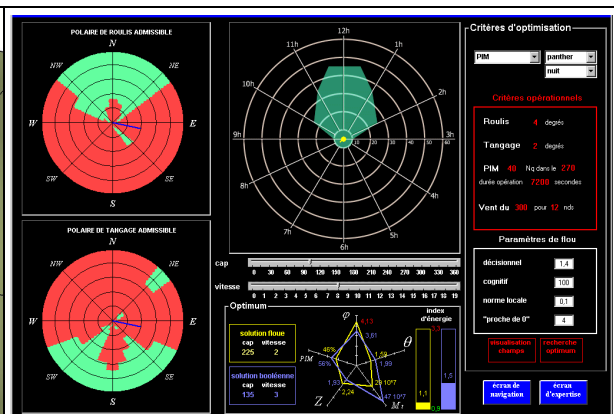


Fig.5: GUI of PrISM tactical routing system, *Capitant and Bertram (2003)*

3. Analysis and Simulation

The word simulation is derived from the Latin word “simulare” which can be translated as “to reproduce”. The VDI (Society of German Engineers) defines the technical term “simulation” as follows: “Simulation is the reproduction of a system with its dynamic processes in a running model to achieve cognition which can be referred to reality”. According to the Oxford dictionary “to simulate” means “to imitate conditions of a situation or process”, specifically “to produce a computer model of a process”. In this sense virtually all computer models used in the design process of ships would qualify as simulations. The field of such computer models in modern shipbuilding processes covers a wide field, including from transport economics, CAD models involving Virtual Reality and ergonomics, finite element analyses, hydrostatic and hydrodynamic analyses, production analyses, etc.

Stability analyses were among the first applications of computers in naval architecture. Today, the naval architect can perform stability analyses in intact and damaged conditions quasi at the push of a button. Two other “classical” applications of computer simulations for ships are CFD (computational fluid dynamics), e.g. *Bertram (2000a)*, and FEA (finite-element analyses). Both have been used for several decades now to support ship design, but today’s CFD and FEA applications are of course far more sophisticated than 20 years ago. In CFD, we see today unsteady viscous flow simulations (albeit with simplified turbulence modeling) with complex two-phase flow simulation (either wave formation including wave breaking or cavitation), e.g. Fig.6. In FEA, unsteady nonlinear analyses are feasible (up to simulations of ships colliding with other ships, rupturing the ship hull). New materials like composites with anisotropic properties, Fig.7, high-frequency analyses (e.g. sound propagation in structures) and fracture propagation involving discontinuous material properties pose new challenges for continued research. Also the combination of hydrodynamics (CFD) and structural mechanics (FEA) in hydro-elastic interaction between fluid and structure is a field of research where many problems are yet to be solved. Hydro-elastic interaction is for example important in determining slamming loads, e.g. *Constantinescu et al. (2004)*.

The last decade has seen a rapid profusion of assorted simulations for ships covering much more than classical hydrostatics, CFD analyses of hull and propeller, or FEA. CFD has been extended to “wind, fire, and ice”:

- Aerodynamic analyses of ship superstructures and ventilated rooms, e.g. Fig.8, *Schmode and Bertram (2002)*,
- simulations of fire in ship rooms, Fig.9, *Junalik et al. (2003)*,
- even numerical ice-breaking, Fig.10.

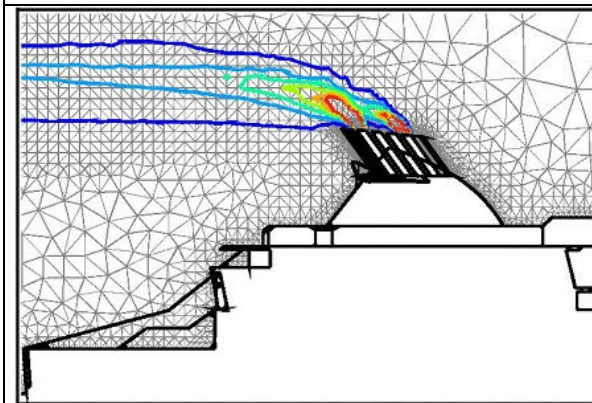
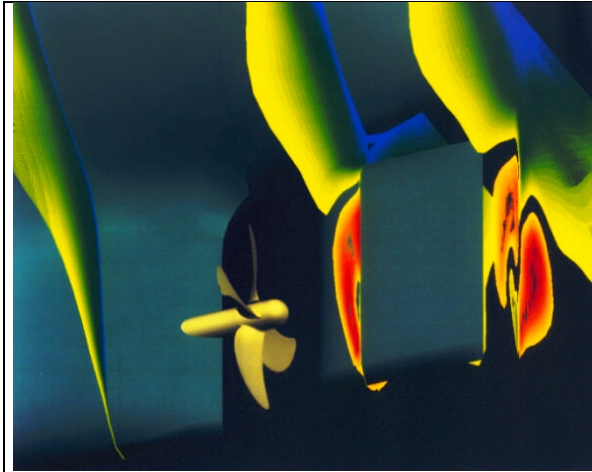


Fig.8: Aerodynamic CFD analysis for fast ship, *Schmode and Bertram (2002)*

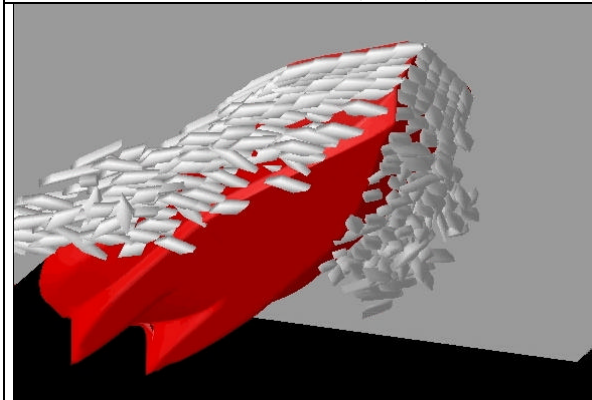


Fig.10: Simulation of ice-breaking at HSVA

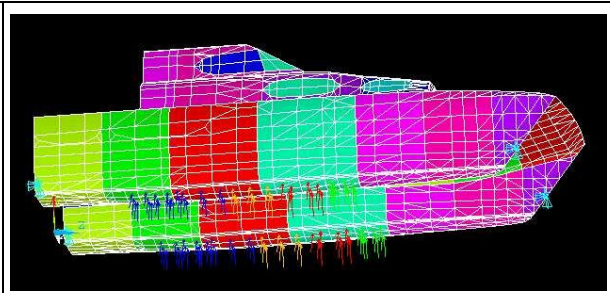


Fig.7: FEA of a catamaran made of composite material, *Luco et al. (2002)*.

← Fig.6: CFD analysis of hull, propeller and rudder, source: Mitsubishi H.I.

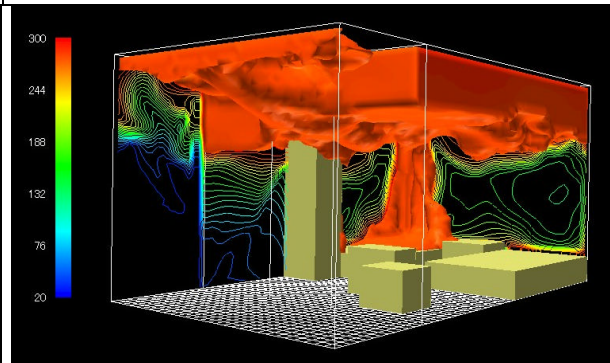


Fig.9: Fire simulation for ship cabin, source: RIAM

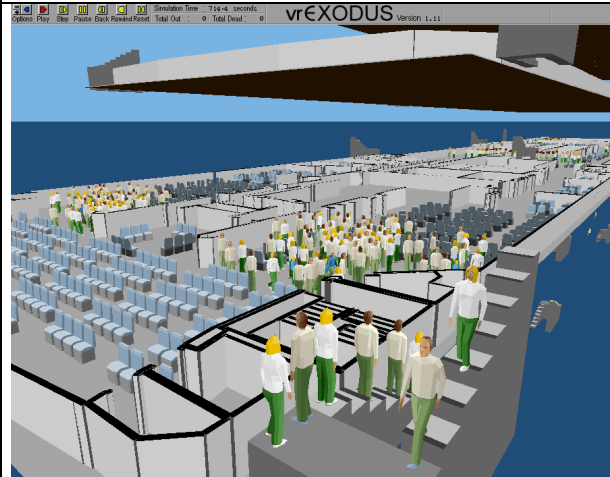


Fig.11: Ship evacuation simulation using DES, *Galea et al. (2003)*

Despite impressive progress, fire simulations for ships face many problems. Classical zonal models are efficient, but questionable in tall rooms with strong 3-d effects. Standard RANSE (Reynolds-averaged Navier-Stokes equations) solvers are not capable to reproduce the evolution of large eddy structures observed in most fire plumes. However, resolution of these eddies in large-eddy simulations (LES) is prohibitively time-consuming. Ship fire simulations face thus the classical dilemma of large complex geometries (if e.g. a whole deck or even several decks are to be simulated) with strong local changes in the simulated properties (temperature, smoke density, etc). Appropriate compromises between available resources and accuracy of the model will change in time as more powerful simulation tools gain practical application maturity.

In the simulation of ice-breaking, we see already many individual (discrete) ice floes in Fig.10. This simulation combines then classical simulation for continuous media (water) with discrete elements. Fire simulation is increasingly also combined with discrete simulation of passengers in evacuation during a fire, e.g. *Galea et al. (2003)*. Discrete event simulation (DES) has evolved rapidly and has proven to be versatile and powerful tool for many applications supporting ship design.

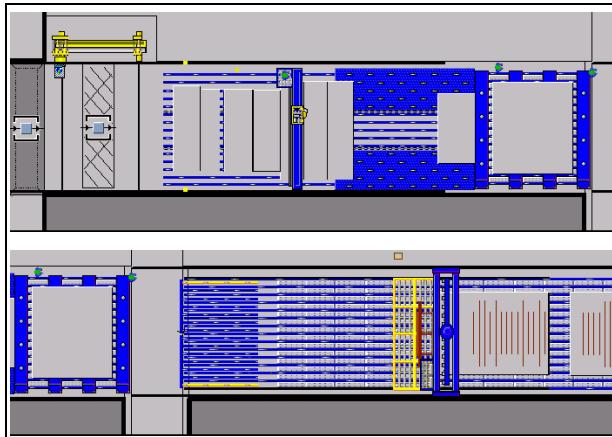


Fig.12: DES for shipyard simulation at FSG,
Steinhauer (2003)

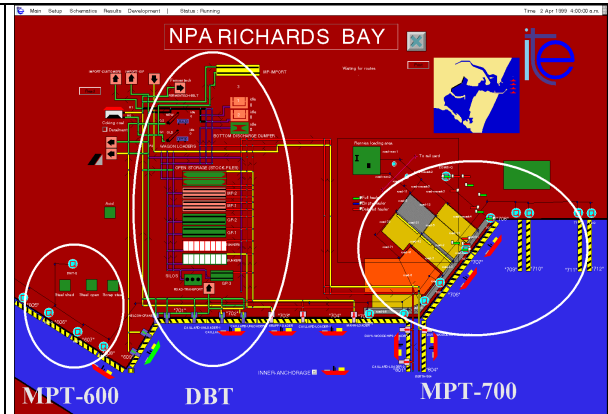
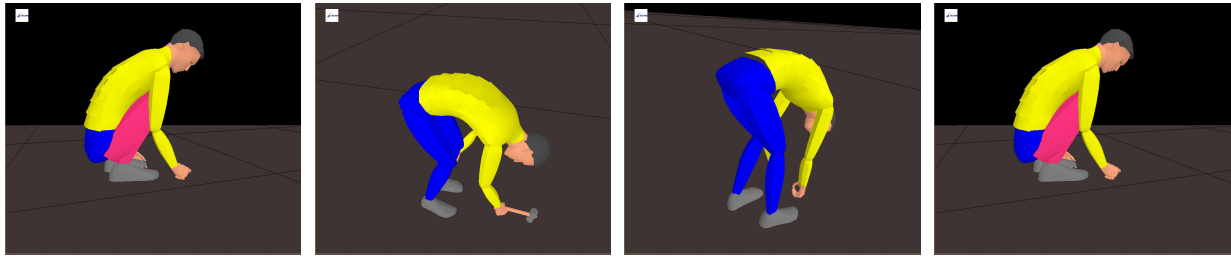


Fig.13: Expert system DES for port simulation,
Simpson et al. (2003)

Discrete event simulation is used in practice in several shipyards to support operational and strategical planning of the shipyard operation, particularly in the assembly-line type of subassembly, e.g. *Steinhauer (2003)*, Fig.12. The actual “theory” behind DES is very simple in comparison to the classical continuous simulation. In DES, individual elements are linked and certain events trigger the execution of other events. Each event is associated with certain properties (attributes). E.g. a stiffener of certain length is welded to a plate with a welding method of certain speed, resulting in a certain time before the welding is finished and the welding head can be moved to another position (again associated with a certain speed). In an object-oriented environment, we have then various objects which can be processed in parallel or in sequence, depending on certain conditions. The logic of the set-up resembles a critical path network plan. The difficulty is rather in realizing the individual elements, how they interact and what their interdependences are, and setting up a model of suitable level of detail (just as detailed as necessary) in acceptable time. Despite a relative simplicity of the fundamental theory, simulation for complex systems offers considerable benefits in qualitative insight (e.g. identifying bottlenecks) and quantitative information (times to perform an assembly or to unload a ship; occupancy rates for workstation; etc.) As often in simulation, the crucial problem involving the most skill consists in setting up the actual model with all necessary attributes.

Simulation is particularly difficult when it involves also humans. This is for example the case in ship evacuation simulation. The trend is here to equip each simulated human with a certain perception and reasoning capability. Such multi-agent systems are subject to research and likely to become increasingly important for a variety of simulations with relevance to ship design. Human decision play also a role in many scheduling tasks. E.g. *Simpson et al. (2003)* present a DES system combined with expert system elements which simulate the heuristic rules employed by port management in assigning ships to berths, Fig.13. The application of *Simpson et al. (2003)* is limited to model a single port, but extension to multi-port simulation with the traffic in between and a limited model of the train interface to the hinterland is envisioned.

Simulation may also be combined with virtual reality (see Chapter 4) if ergonomics are important. Virtual mannequins can then be studied in performing tasks in the building or operation of ships. *Sasaki (2003)* presents e.g. such a combination of shipyard simulation with virtual reality simulation of certain assembly tasks, Fig.14. Similarly, virtual reality mannequins have been used to study feasibility of torpedo loading in confined submarines and crew operation in engine rooms.



Set the small parts fixing the parts position check the parts position welding

Fig.14: Examples of mannequin simulating basic tasks in ship sub-assembly, *Sasaki (2003)*

4. Visualization

The beginnings of using ship lines as an aid in ship design date back at least to the early 17th century. Rembrandt's (1606-1669) painting "The shipbuilder and his wife" shows the naval architect Jan Rijksen, chief designer for the Dutch East India company in the early 17th century. Jan Rijksen is depicted at work, drawing cross sections and a keel with compass and ruler, Fig.15. *Chapman (1775)* mentions already the use of a family of parabolas for waterlines and other ship curves. Some of the traditions of lines displays have in essence been kept since the time of Rembrandt's painting.



Fig.15: Rembrandt's "The shipbuilder and his wife"



Fig.16: Cross sections and splines (historical)



Fig.17: Ducks and wooden spline, TU Berlin

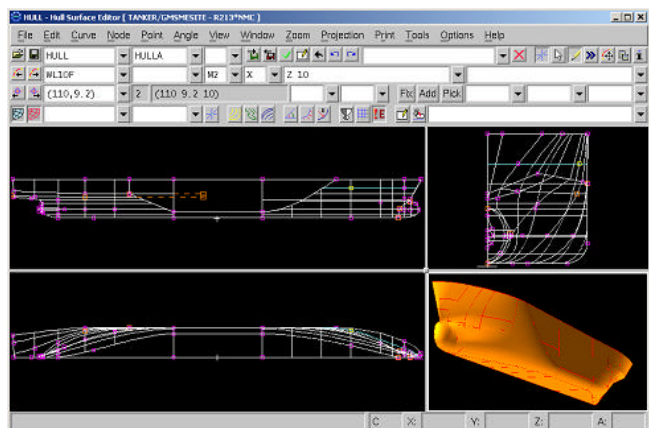


Fig.18: Modern CAD description, NAPA Oy

Technical drawings based on classical descriptive geometry were used already in the 15th century. Initially, compass and straight ruler were the only tools used, limiting forms to straight lines and

(small) parts of circles. The limitation of tools to describe form elements limited the potential designs. Wooden ships of the 16th century used already the ship definition based on cross sections erected on the longitudinal keel. The hull planks were aligned along thin wooden splines over the cross sections, Fig.16. These splines can be regarded as the forefathers of the splines used later on drawing tables to create ship lines on paper, Fig.17, and the mathematical splines used in CAD (computer aided design) programs, Fig.18. *Harries (1998)* is recommended for a good overview of the state of the art in ship hull design systems.

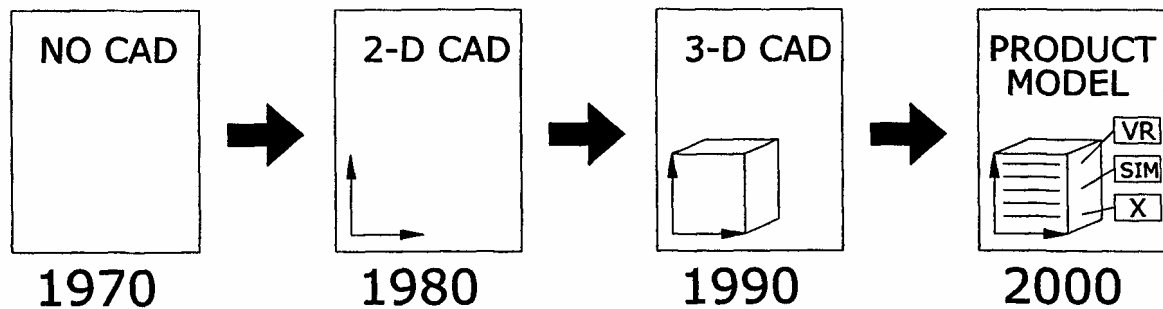


Fig.19: Development of ship CAD systems towards three-dimensional product models, *Aarnio (2000)*

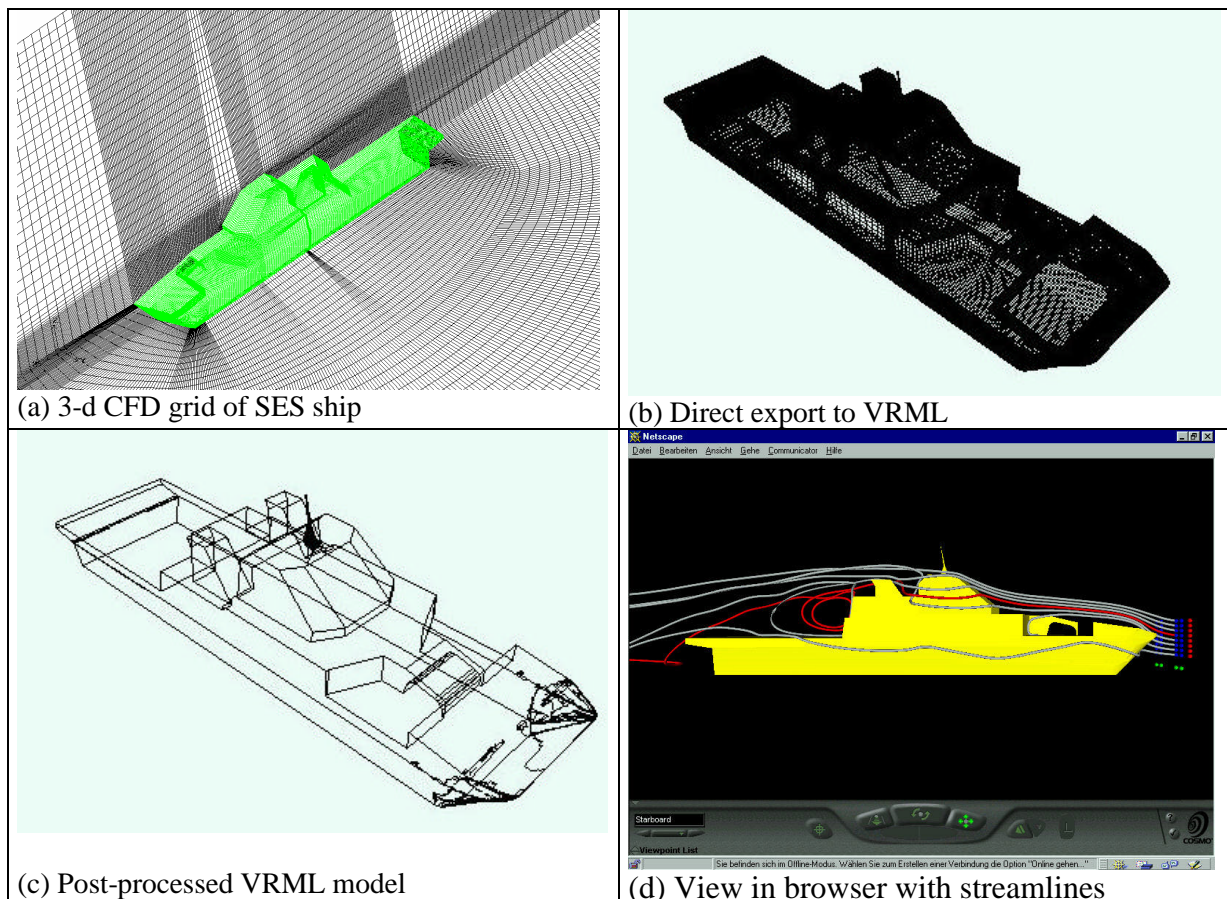


Fig.20: Virtual Reality models offer sometimes better visualization, but require usually some work as direct export from CAD gives formally correct syntax, but not functionality, *Lindenau and Bertram (2003)*,
http://www.ifs.tu-harburg.de/IFS/AB/AB-3-13/Arbeitsschwerpunkte/VRML_HTML/vrml_main.htm

While we retained essentially the same representation of ships over centuries, the progress in CAD systems towards three-dimensional product data models, Fig.19, allows today not only to perform a large variety of analyses and simulations (as discussed in chapter 3), but also advanced representation in photo-realistic displays or in Virtual Reality. While better visualization options are a bonus, but the

main benefit of 3-d design lies in data management and advanced simulation options supporting first-principle design.

The Virtual Reality Modeling Language (VRML) allows to model arbitrary geometries (like ships) consisting of polygons. The language is particularly suited for the internet and has become ISO standard in its 1997 version. A VRML description allows to chose perspective and zoom at will. Public-domain browsers allow viewing the objects with walk-through or fly-through capabilities. The resulting files are often very small (typically less than 1 Mbyte), while offering high resolution. Many CAD programs allow export to VRML format and VRML objects can be found on the internet and integrated to new “worlds”. However, direct export of CAD descriptions or simulation models (grids) for other purposes to VRML results generally in unsuitable models. While formally following the VRML syntax, these directly exported models are typically far too detailed and big for the purposes of Virtual Reality viewing and require post-processing, Fig.20, as described e.g. by *Lindenau and Bertram (2003)*, *Wauchope et al. (2003)*.

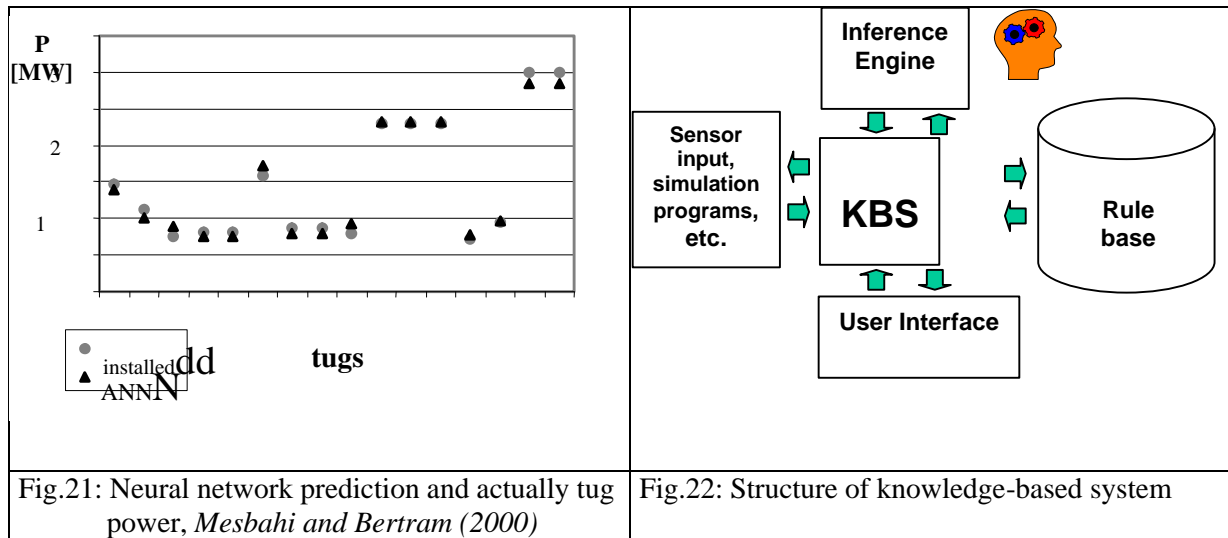
5. Knowledge and Decision

Knowledge remains at the heart of design. Knowledge comes largely from education and work experience. However, machine learning may on occasion support human knowledge. Artificial neural networks (ANNs), *Mesbahi (2003)*, allow very general functional fitting to data sets. They can be used similarly as splines for curve fitting, but also allow to derive functional relations between input vectors and output vectors. This is useful for some applications in ship design. ANNs can e.g. be used to determine trends or first estimates for certain ship types, Fig.21, e.g. *Bertram and Mesbahi (2000)*, *Mesbahi and Bertram (2000)*, enhance numerical CFD simulation with empirical corrections, e.g. *Salas et al. (2002)*, or be used in filtering data in ship operation, e.g. for maneuvering trials, *Hess and Faller (2000)*, etc. ANNs can be applied if the functional relation between data is unknown or too complex to be explicitly specified. However, as with most AI techniques, ANNs benefit from adding human knowledge. If human knowledge can e.g. eliminate irrelevant parameters or supply already a good approximation e.g. from theoretical analysis, the ANN will perform that much better in approximating the residual quantities. ANNs are also successfully combined with optimization techniques where the ANN approximates Pareto curve (trends for curves which represent optima one parameter for fixed other parameters), usually making the global search considerably faster, *Bertram (2003a)*.

Knowledge-based systems (KBS) have been successfully applied to a wide variety of marine applications, most of them concerning ship operation, *Kaeding and Bertram (1996)*, *Bertram (2000b)*. KBS differ from traditional data-processing computer programs in their method of operation. Conventional programs are optimized for numeric-processing, whereas the knowledge-based system concentrates on the representation and manipulation of information as symbols. Another noteworthy feature of KBS is their suitability for large and complex problem solution characterized by inexact, incomplete and uncertain information. Their structure includes an explicit body of embedded knowledge and a separate, identifiable inference mechanism, Fig.22.

Knowledge-based systems support best tasks that are relatively simple and do not involve common sense or a “wide picture”. A classical example is collision avoidance of ships, *Kasai and Bertram (1996)*, where relatively few traffic rules and heuristic rules of experienced mariners suffice to get a performance equal or superior to humans. Ship design on the other hand, is a largely unstructured and creative task, making knowledge-based systems rather difficult. However, partial support can be given and aid the designer in practice as demonstrated by the Dutch navy in applying QUAESTOR, the hydrodynamics KBS of MARIN, *van Es and van Hees (2003)*. Propeller design is far more structured than ship design, making KBS supported propeller design an attractive option, *Reich et al. (1997)*. Similarly, complex simulation tools supporting ship design as described in chapter 3 can be enhanced by KBS technology. Input generation for CFD analyses can be effectively accelerated and simplified using this approach, *Marzi et al. (1997)*. Also KBS can simulate human decisions in scheduling and operational simulation, *Simpson et al. (2003)*. KBS can also support optimization, either in supplying

suitable constraints, e.g. as in the routing application of *Capitant and Bertram (2003)*, in selecting optimization control parameters or in finding reasonably good solutions as starting points for a formal optimization as reviewed by *Bertram (2003a)*.



6. Conclusion

Design combines the elements high-lighted above: generation, analysis, visualization and knowledge. None of these should be seen by themselves. Only in proper combination is the full potential for advanced designs revealed. The individual chapters showed already on occasion the general trend towards combination of techniques. This trend will continue: simulation tools with Virtual Reality displays, knowledge based systems with simulation tools, simulation training neural networks, etc. Stand-alone techniques have reached a high degree of maturity and further progress is best obtained by properly combining techniques. This requires often interdisciplinary cooperation and funding frameworks as well as modern communication possibilities allow increasingly to source the best partners worldwide for a given problem. The challenge of the future lies in using these opportunities and training new generations of engineers to properly master these tools with creativity, critical distance and an ethical conscience.

References

- AARNIO, M. (2000), *Early 3-d ship model enables new design principles and simulations*, 1st Int. Conf. Computer and IT Applic. Mar. Industries, COMPIT, Potsdam, pp.5-17
- ABT, C.; HARRIES, S.; HEIMANN, J.; WINTER, H. (2003), *From redesign to optimal hull lines by means of parametric modelling*, 2nd Int. Conf. Computer and IT Applic. Mar. Industries, COMPIT, Hamburg, pp.444-458
- ANDREWS, D. (1981), *Creative ship design*, Trans. RINA 123, pp.447-471
- BERTRAM, V. (2000a), *Practical ship hydrodynamics*, Butterworth-Heinemann, Oxford
- BERTRAM, V. (2000b), *Knowledge-based systems for ship design and ship operation*, 1st Int. Conf. Computer and IT Applic. Mar. Industries, COMPIT, Potsdam, pp.63-71
- BERTRAM, V. (2003a), *Optimization in ship design*, 39th WEGEMT School OPTIMISTIC – Optimization in Marine Design, Mensch&Buch Verlag, pp.27-52
- BERTRAM, V. (2003b), *Cyber-ships – Science Fiction and reality*, 2nd Int. Conf. Computer and IT

Applic. Mar. Industries, COMPIT, Hamburg, pp.336-349

BERTRAM, V.; EL MOCTAR, O. (2002), *Design for production in ship hull design*, 37th WEGEMT School “Advanced Shipbuilding and Shipping”, Madrid

BERTRAM, V.; ISENSEE, J.; GUDENSCHWAGER, H. (1998), *Application of optimization shells to design problems*, Ship Technology Research 45, pp.88-95

BERTRAM, V., MESBAHI, E. (2000), *Applications of neural nets in ship design*, Jahrbuch der Schiffbautechnischen Gesellschaft 94, Springer, pp. 207-211

BUTRUILLE, A. (2002), *Modélisation paramétrique des formes de coques pour l’optimisation*, Diploma thesis ENSIETA

CAPITANT, C.; BERTRAM, V. (2003), *Künstliche Intelligenz für kurzfristige Routenplanung*, Hansa 140/12, pp.22-23

CHAPMAN, F.H. (1775), *Architectura Navalis Mercatoria*, Reprint Sheridan House (1901)

CONSTANTINESCU, A.; NEME, A.; BERTRAM, V.; DOUTRELEAU, Y.; MOYNE, S.; PESEUX, B. (2004), *Finite elements simulations of dihedral-shape and conical-shape shell structures slamming*, submitted to 8th Int. Conf. Flow-Induced Vibrations, FIV2004, Paris

DUVIGNEAU, R.; VISONNEAU, M. (2003), *Shape optimization strategies for complex applications in computational fluid dynamics*, 2nd Int. Conf. Computer and IT Applic. Mar. Industries, COMPIT, Hamburg, pp.154-161

ERIKSTAD, S.O. (1996), *A decision support model for preliminary ship design*, Ph.D. thesis, Univ. of Trondheim

EVANS, J.H. (1959), *Basic design concepts*, ASNE Journal

GALEA, E.R.; GWYNNE, S.; LAWRENCE, P.J.; BLACKSHIELDS, D.; EWER, J.; WANG, Z.Z.; HURST, N.; MAWHINNEY, N. (2003), *The application of fire and evacuation simulation in ship design*, 2nd Int. Conf. Computer and IT Applic. Mar. Industries, COMPIT, Hamburg, pp.55-69

GUDENSCHWAGER, H. (2003), *Application of optimization in basic ship design*, 39th WEGEMT School OPTIMISTIC – Optimization in Marine Design, Mensch&Buch Verlag, pp.173-190

HARRIES, S. (1998), *Parametric design and hydrodynamic optimization of ship hull forms*, Mensch&Buch Verlag, Berlin

HESS, D.; FALLER, W. (2000), *Simulation of ship manoeuvres using recursive neural networks*, 23th Symp. Naval Hydrodyn., Val de Reuil

ISENSEE, J.; BERTRAM, V. (2004), *Quantifying external costs of emissions due to ship operation*, submitted to J. Engineering Maritime Environment

JUNALIK, B.; BERTRAM, V.; EL MOCTAR, O. (2003), *Preliminary investigations for CFD fire simulations in ship rooms*, 6th Numerical Towing Tank Symposium, Rome

KAEDING, P.; BERTRAM, V. (1996), *Artificial intelligence for ship automation -- Technical and economical aspects of reduced crews*, IfS Report 572, Univ. Hamburg

- KASAI, H.; BERTRAM, V. (1996), *Artificial intelligence for ship operation control Part I: Automatic navigation including collision avoidance*, Hansa 133/5, pp.18-21
- KOUSHAN, K.; STEEN, S.; ZHAO, R. (2002), *Wake wash minimization by hull form optimization using artificial intelligence*, 3rd Int. Conf. High-Performance Marine Vehicles, HIPER, Bergen, pp.262-273
- LINDENAU, O.; BERTRAM, V. (2003), *The making of a VRML model for an SES with streamlines and pressure distribution*, 2nd Int. Conf. Computer and IT Applic. Mar. Industries, COMPIT, Hamburg, pp.5-14
- LUCO, R.; SALAS, M.; TARFAOUI, M.; BERTRAM, V. (2002), *Finite element structural modeling of a composite-material multihull*, 3rd Int. Conf. High-Performance Marine Vehicles, HIPER, Bergen, pp.297-305
- MARZI, J.; BERTRAM, V.; STÖHRMANN, H. (1997), *User-friendly hydro-numeric hull design by incorporation of expert knowledge*, ICCAS'97, Yokohama, pp.351-361
- MESBAHI, E. (2003), *Artificial neural networks – Fundamentals*, 39th WEGEMT School OPTIMISTIC – Optimization in Marine Design, Mensch&Buch Verlag, pp.191-216
- MESBAHI, E.; BERTRAM, V. (2000), *Empirical design formulae using artificial neural networks*, 1st Int. Conf. Computer and IT Applic. Mar. Industries, COMPIT, Potsdam, pp.292-301
- MISTREE, F.; SMITH, W.F.; BRAS, B.; ALLEN, J.K.; MUSTER, D. (1990), *Decision-based design: A contemporary paradigm for ship design*, Trans. SNAME 98, pp.565-595
- REICH, Y.; BERTRAM, V.; FRIESCH, J. (1997), *The development of a decision support system for propeller design*, ICCAS'97, Yokohama, pp.363-378
- RIGO, P. (2003), *An integrated software for scantling optimization and least production cost*, Ship Technology Research 50, pp.125-140
- SCHMODE, D.; BERTRAM, V. (2002), *Aerodynamic flow computations for a Superfast ferry*, 3rd Int. Conf. High-Performance Marine Vehicles, HIPER, Bergen, pp.345-354
- SALAS, M.; MESBAHI, E.; BRINK, K.E.; BERTRAM, V. (2002), *Empirical roll damping formula derived by adaptive neural network application*, Schiff+Hafen 54/2, pp.56-57
- SASAKI, Y. (2003), *Application of factory simulation to the shipyard*, 2nd Int. Conf. Computer and IT Applic. Mar. Industries, COMPIT, Hamburg, pp.362-376
- SIMPSON, D.; FURSTENBERG, L.; KALATHAS, E.; BERTRAM, V. (2003), *Virtual prototyping for developing South African ports*, 2nd Int. Conf. Computer and IT Applic. Mar. Industries, COMPIT, Hamburg, pp.235-246
- STEINHAUER, D. (2003), *The virtual shipyard – Simulation in production and logistics at Flensburg*, 2nd Int. Conf. Computer and IT Applic. Mar. Industries, COMPIT, Hamburg, pp.203-209
- VAN ES, K.; VAN HEES, M.T. (2003), *Application of knowledge management in conceptual naval ship design*, 2nd Int. Conf. Computer and IT Applic. Mar. Industries, COMPIT, Hamburg, pp.350-362
- WAUCHOPE, K.; EVERETT, S.; TATE, D.; MANEY, T. (2003), *Speech-interactive virtual environments for ship familiarization*, 2nd Int. Conf. Computer and IT Applic. Mar. Industries, COMPIT, Hamburg, pp.70-83

Use of Genetic Algorithms in Propeller Foil Design

Marco Ferrando, DINAV, Genova/Italy, ferrando@dinav.unige.it

Andrea Lommi, CETENA, Genova/Italy, andrea.lommi@cetena.it

Antonio Traverso, CETENA, Genova/Italy, antonio.traverso@cetena.it

Abstract

One of the most popular tools for design of 2-D foil sections is that developed by Prof. Eppler. Unfortunately the required input data are not correlated to the actual shape of the profile in a straightforward way. A mathematical solution to the problem of 2-D foil sections design with arbitrary constraints does not exist, and available design tools (like the Eppler code) do not allow a real optimization of produced solutions, but rather a simple evaluation, with the result that if contrasting characteristics have to be jointly optimized the optimal trade-off is difficult to devise. Depending on the chosen design objective, like for instance the optimization of a cavitation bucket, deterministic optimizers like the gradient method could get stuck in local minima or could even be not exploitable, due to the non-linearity of the search space and therefore a use of a stochastic method is necessary. During the past years, Genetic Algorithms have been applied on a variety of engineering problems, showing a high robustness and reliability on handling non-linear and sub-optimal solutions-filled search spaces. This paper describes the development of a tool for the design of foil shapes that integrates a genetic algorithm, a revised Eppler code and a procedure to compare foil sections having different zero-lift angles. Results are evaluated by means of an objective function that can be adapted to the goal for any project and, in this case, an example is given by optimising a cavitation bucket.

1. Introduction

In the last decade traditional approach to propeller design has evolved side by side with new techniques of optimization that moved from the classical numerical approach based on numerical methods toward the field of Genetic Algorithms (GA) which seem to be suitable for wider ranges of application than those based on analytical theory of optimization.

In propeller design the starting point is the airfoil section and, since their introduction, NACA Mod 66 and NACA 16 profiles have been the most used blade sections and considered as a good starting point for marine propeller design. Anyway as cavitation phenomena are involved, modification of geometrical parameter such as rake and skew are not sufficient to ensure a complete control of propeller performances and a new starting point might be needed, and this can be a complete different choice of blade sections. Unfortunately propeller operates in a fluid with a non uniform field of velocity so that each blade section operates in different condition with different angle of attack, which in turns changes as the propeller rotates. This makes impossible to design an optimized propeller for the whole range of operation, but it is still possible to choose one or two blade section, which are considered the most important, and try to optimize them.

Airfoil design can be achieved with several codes, such as the one developed by *Eppler and Somers (1980)*. The best choice of input parameter is defined and found with a Genetic Algorithm that takes into account the previously mentioned points by acting on the characteristics of the airfoil for which a suitable coding of the parameters has been defined. The obtained airfoil configurations represent optimality-close tradeoffs between the required characteristics.

2. The Eppler Code

The code developed by Richard Eppler, *Eppler and Somers (1980)*, uses the potential flow model to generate the geometry of an airfoil which satisfies a certain set of condition imposed by the designer. The input for this program is extremely complex as the flow field around the airfoil is not described either by a velocity or a pressure field.

The approach followed by this code is based on a conformal mapping of the airfoil on the complex plane ζ . Airfoil points are mapped over a unit circle and each point is identified by an angle ϕ . Profile outline is dividend into N_i segments, the requirements to design the airfoil is that over each segment for $2 < i < N-1$, there is an incidence angle α^*_i which ensures a constant distribution of velocity over the segment. So arc percentages and angles of incidence are the main input data for Eppler code. Program output comprehends the complete description of the profile performances, including a cavitation bucket.

The following two lines are the input data that are used by the code to obtain the foil section illustrated in Fig.1

```
TRA1  502 58.1 4.0 0 4.25 63.0 -.21 120 -.21
TRA2  502 4.0 12.0 2 .5 .768 4.0 52.5 2 .5 .768 3 1.0 0 0
```

These data specify the length of the closure regions 1 and 8, of the pressure recovery regions of the suction side 2 and of the pressure side 7 and the regions 3, 4 5 and 6 where the velocity is to be constant at specified angles of attack.

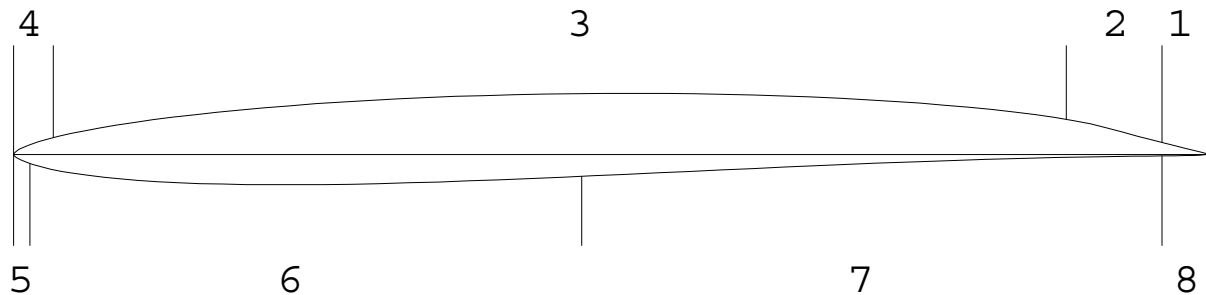


Fig.1: Example of profile regions to be specified in the input data.

Considering that there are some more minor parameters to handle, it appears that it's very difficult to find an intuitive correlation between input data and resulting profile shape and performances. Notwithstanding the input difficulty, the Eppler's code has been widely used and other authors used different methods to overcome input problems, *Kuiper and Jessup (1993)*.

For this reason it was decided to use a GA to drive the code to search an optimum profile. In this case the goal was to design a foil section with better performances with respect to cavitation comparing with a traditional NACA section of prescribed thickness. It is obvious that the objective function of the optimization process must include some type of form control since the Eppler's code can produce peculiar shapes as the one illustrated in Fig.2.

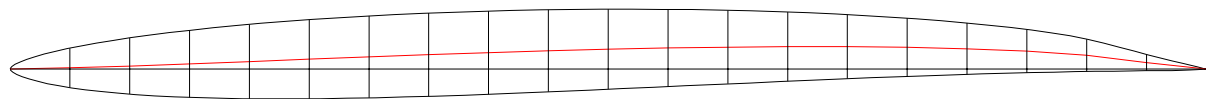


Fig.2: S shaped camber line.

This profile can be difficult to use when designing a propeller, because of its S shaped camber line.

A first attempt was done with a GA found on Internet, the so called Simulated Annealing Method, *Kirkpatrick et al. (1983)*, in order to test the feasibility of the approach and the choice of the fitness function, but it was clear from the first runs that constraints over input were too strong to be handled by a generic routine and a complete control over the "population choice" was needed in order to prevent Eppler code from crashing. That is why a home made GA was developed and used for this task.

3. Fitness Function Formulation

The fitness function represents a mathematical formulation of the evaluation criteria used to assess the quality of the results. Its formulation and use in the optimization process is a compulsory step for devising optimized solutions. Generally for airfoil design drag and lift coefficient are two of the most important parameters to take into account, but this was not our goal at the moment, as we were more interested in cavitation the constraints were well defined: we needed a cavitation bucket large enough to hold all the working points.

A great number of functions was tried, aiming to achieve a bucket as large or as squared as possible, but the results were still not satisfactory. As we wanted all the points to be held inside the bucket, we defined a fitness function whose value increased each time a point was inside. Furthermore, as the importance of each point was related to its pressure coefficient and the lower is this coefficient the more important is the point, we chose the following fitness function:

$$F = \sum_i \frac{\delta_b(d)}{C_p^i} G(s) \quad (3.1)$$

where the sum is over the working points δ_b is 1 if the point is inside the bucket, otherwise:

$$\delta_b(d) = \frac{e - \left(1 + \frac{1}{d}\right)^d}{e - 1} \quad (3.2)$$

where d is the distance from the border of the bucket. This function was introduced not to rise the penalty of airfoil whose bucket was very close to the working points but not large enough to hold them all. Finally, as thickness is a mandatory requirement in airfoil design in order to ensure structural strength, fitness function was multiplied by a narrow band Gaussian function peaked around the required thickness $s_{req.}$, expressed by:

$$G(s) \propto e^{-\left(s - s_{req.}\right)^2} \quad (3.3)$$

The resulting fitness function has been used for ranking the chromosomes in the genetic process described in the next paragraph.

4. The Genetic Algorithm

Genetic Algorithms are versatile optimization algorithms based on the theory of evolution *Holland (1975)*. Standard implementations, *Goldberg (1989)*, code a population of P trial solutions into binary strings called chromosomes; as the evolutive process goes on, the population characteristics improve thanks to the crossover, reproduction and mutation operators applied on chromosomes, selected according to the value of a suitable fitness function. As a consequence, the best solutions spread their characteristics among the population, causing the fitness function calculated on the basis of population's chromosomes to increase; once it has reached a fixed threshold level or a maximum number of generations has been completed, the procedure is stopped.

As far as our implementation is concerned, the generic trial solution on whose parameters the GA acts is represented by the following expression:

$$\Phi^{trial} = \{(\sigma_i, \theta_i), 0 \leq i \leq B-1\} \quad (4.1)$$

where the couple (σ_i, θ_i) contains the i^{th} segment and angle of the considered trial solution, and B is the total number of couples needed for describing the profile. Moreover, for avoiding the production of non-feasible solutions, constraints are imposed on the chromosomes parameters, and all of the P chromosomes are checked before calculating their fitness value. The chosen constraints can be expressed by these expressions:

$$\begin{aligned}\sigma_i &\in [\alpha_{\min}, \alpha_{\max}] \\ \theta_i &\in [\theta_{\min}, \theta_{\max}]\end{aligned}\quad (4.2)$$

being (α_l, α_u) and (θ_l, θ_u) the couples describing the lower and upper bound for α and θ parameters, respectively.

Unlike standard GAs, we directly act on the real parameters of the profile for avoiding time-consuming coding/decoding procedures and quantization errors deriving from expressing real numbers with boolean words.

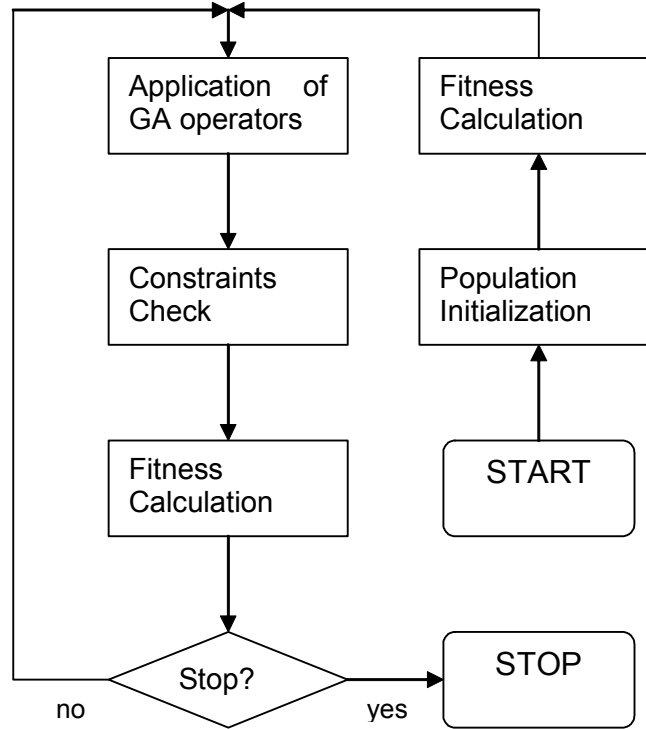


Fig.3: Flowchart of the Genetic Algorithm

As usual, for the adaptation of trial solutions over the environment, chromosomes are ranked according to their respective value of the fitness function expressed in formula (3.1) before genetic operators are applied.

As far as the GA operators are concerned, crossover is a standard single-point one that is applied on couples of roulette wheel-selected, *Goldberg (1989)*, chromosomes with probability ζ_C , and mutation alters each gene of every chromosome with probability ζ_M by perturbing its value in the range expressed by:

$$\begin{aligned}\alpha_i^{new} &= rand(\alpha_i^{old} - \mu_\alpha, \alpha_i^{old} + \mu_\alpha) \\ \theta_i^{new} &= rand(\theta_i^{old} - \mu_\theta, \theta_i^{old} + \mu_\theta)\end{aligned}\quad (4.6)$$

where μ_α and μ_θ are the chosen mutation intervals for α and θ parameters, respectively, usually set to a quarter of the whole range.

The flowchart of this procedure is shown in Fig.3.

5. Profile Comparison

The procedure for comparing the performances of new profiles with the original one checks the position of the operating points with respect to the cavitation bucket.

Operating points of the original profile are obtained by the lifting surface code that analyses the working conditions of each blade section that, during a complete revolution, meets different inflow conditions due to the non homogeneous wake.

The starting data set is then comprised of the original cavitation bucket and of the operating points, as illustrated in the following Fig.4.

Since the input data of Eppler's code do not allow a direct control of the zero lift angle of the foil, the new profiles can have a considerably different zero lift angle from the original one to which the operating points refer. For this reason the original operating points can not be superimposed on the cavitation bucket of the new profiles.

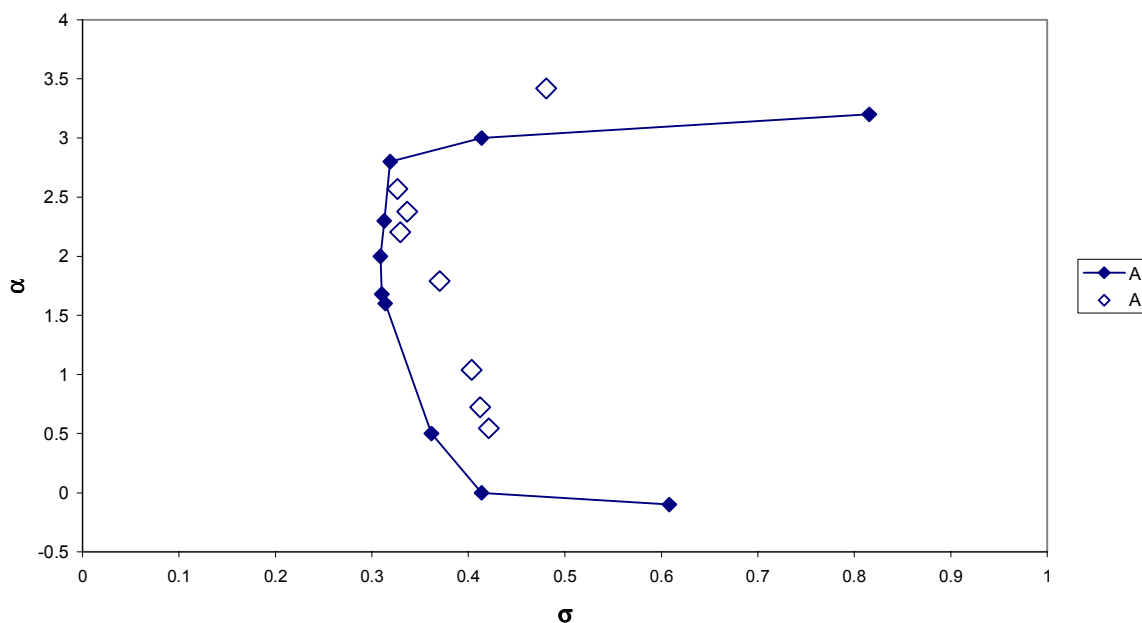


Fig.4: Cavitation bucket and operating points

The fundamental design constraint of an improved propeller design is usually the thrust value. In order to produce the same thrust, if the new blade sections have different zero lift angles they should also have different incidence angles.

With reference to the symbols defined in Fig.5 and assuming that the following relations holds:

$$C_L = 2\pi\alpha_i$$

to produce the same lift two different foils, at the design point, should have the same hydrodynamic angle of incidence α_i .


$$\begin{cases} \alpha_I = \alpha_0 + \alpha \\ \alpha_I = \alpha_{01} + \alpha_1 \end{cases}$$
$$\begin{cases} \alpha = \alpha_I - \alpha_0 \\ \alpha_1 = \alpha_I - \alpha_{01} \end{cases}$$
$$\begin{cases} \varphi = \alpha + \beta_I \\ \varphi_1 = \alpha_1 + \beta_I \end{cases}$$

22

From another point of view and under the same assumptions, the incidence angles of the new operating points can be obtained from the following equation:

$$\alpha_0 + \alpha = \alpha_{01} + \alpha_1$$

that implies:

$$\alpha_{1i} = \alpha_i + (\alpha_0 - \alpha_{01})$$

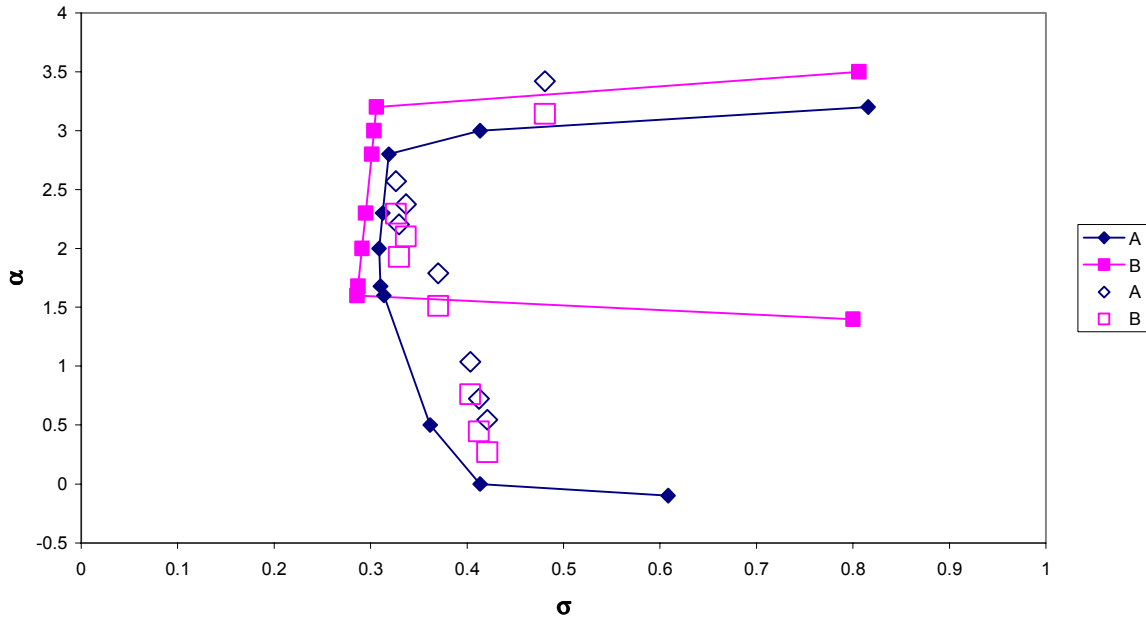


Fig.6: Profiles performance comparison

The incidence angles of the new operating points differ from those of the original profile by:

$$(\alpha_0 - \alpha_{01})$$

The result of this operation is illustrated in Fig.6 that compares cavitation buckets and operating points pertaining to two profiles having different zero lift angle.

A procedure was developed to obtain the working points of new profiles in order to perform a correct comparison of the performances.

6. Code performance

The present work originated from the necessity of developing a tool for optimization of foil sections with regard to cavitation. To this effect, a fitness function was designed in order to compare different solutions in terms of cavitation bucket. The computer code was validated on a test case for which a very good solution was already available and used as a benchmark. The code was run and it stopped after 2840 iterations.

As an essential example of the code performances, a basic comparison is given here considering the solutions of the first iteration, of the optimum solution and of the benchmark. Fig.7 illustrates the cavitation buckets and the relevant operating points of the profiles of the first iteration, of the

optimum solution and of the benchmark.

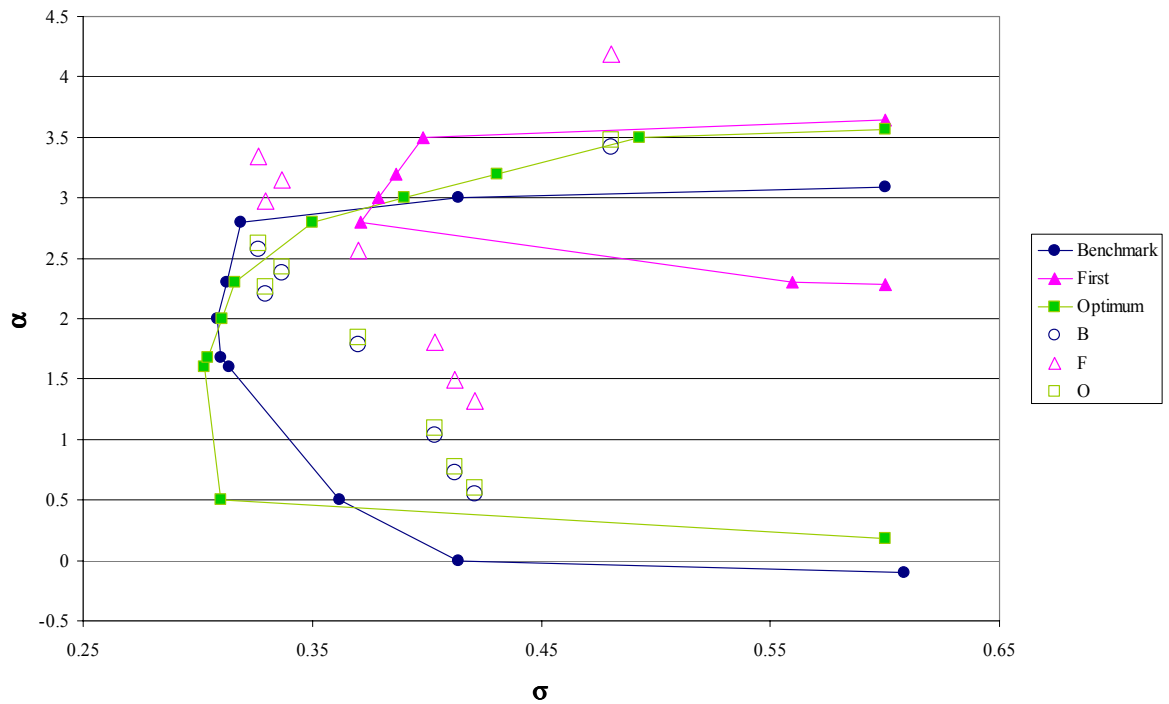


Fig.7: Benchmark results – cavitation buckets comparison

The performance of the first iteration is clearly totally unacceptable since none of the operating points falls inside the relevant cavitation bucket. The optimum solution compares very well with the benchmark. The optimum section is better than the benchmark in terms of laminar cavitation since its bucket comprehends a working point that is out of the bucket of the benchmark. Conversely the optimum section is slightly poorer than the benchmark in terms of bubble cavitation since one of the operating points is outside its bucket.

Rigorously speaking, the benchmark section is better than the optimum since bubble cavitation is worse than laminar cavitation. Considering the fitness function that was used for the optimization, the optimum section is better than the benchmark since the external point is nearer to the bucket boundary. This problem can easily be solved adding some penalty to the points that are out of the bucket in the region of bubble cavitation. Fig.8 reproduces the shape of the considered sections. As it can be observed, the geometry of the optimum solution is similar to that of the benchmark; notwithstanding the fact that the fitness function did not include any control on the section shape other than the thickness, the algorithm does not produce “creative” solutions.

The code performs very well also increasing the number of parameters of the problem. Figs.9 and 10 compare between optimum solutions obtained with four and six pairs of parameters (arc percentage, angle).

7. Results and conclusions

The use of a code which requires complex input takes a lot of time and several calculations must be attempted reach the optimum result. Besides, input data need to be checked prevent the code from crashing. Computer programs such as Eppler’s are not user friendly and require considerable experience by the user to get meaningful results, since input parameters are not directly correlated with the shape of the resultant foil section. For this reason a high effort is required from the user if the design of a high performance profile is attempted.

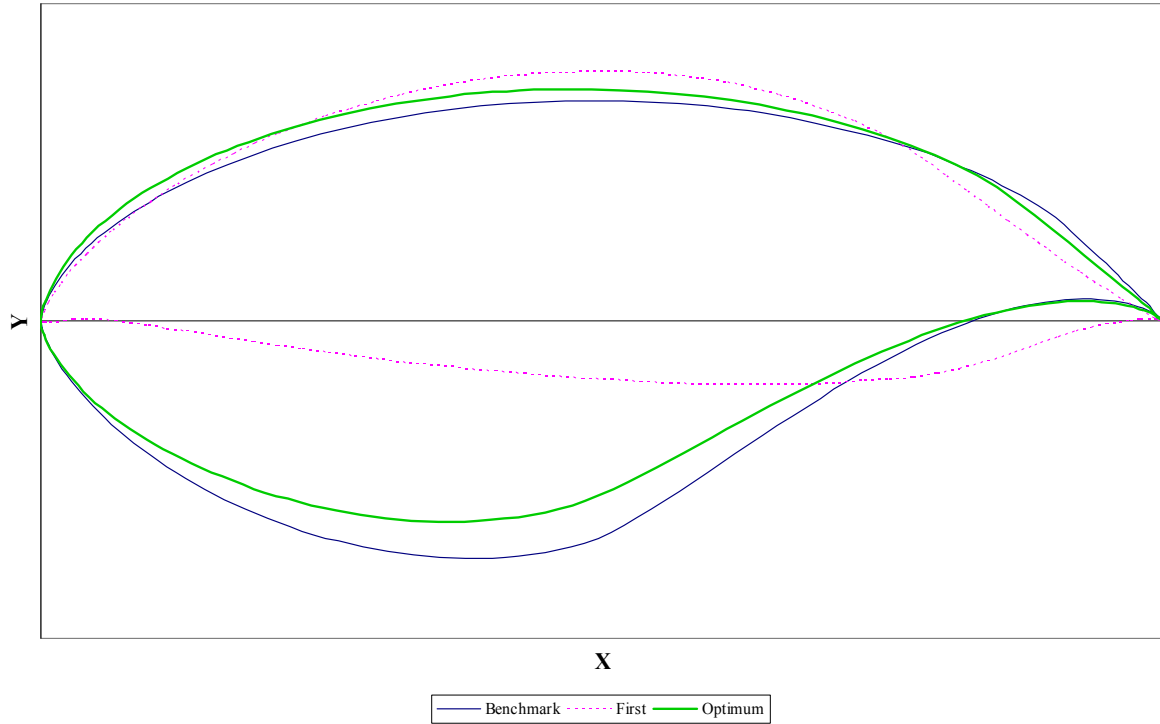


Fig.8: Benchmark results – foil section geometry

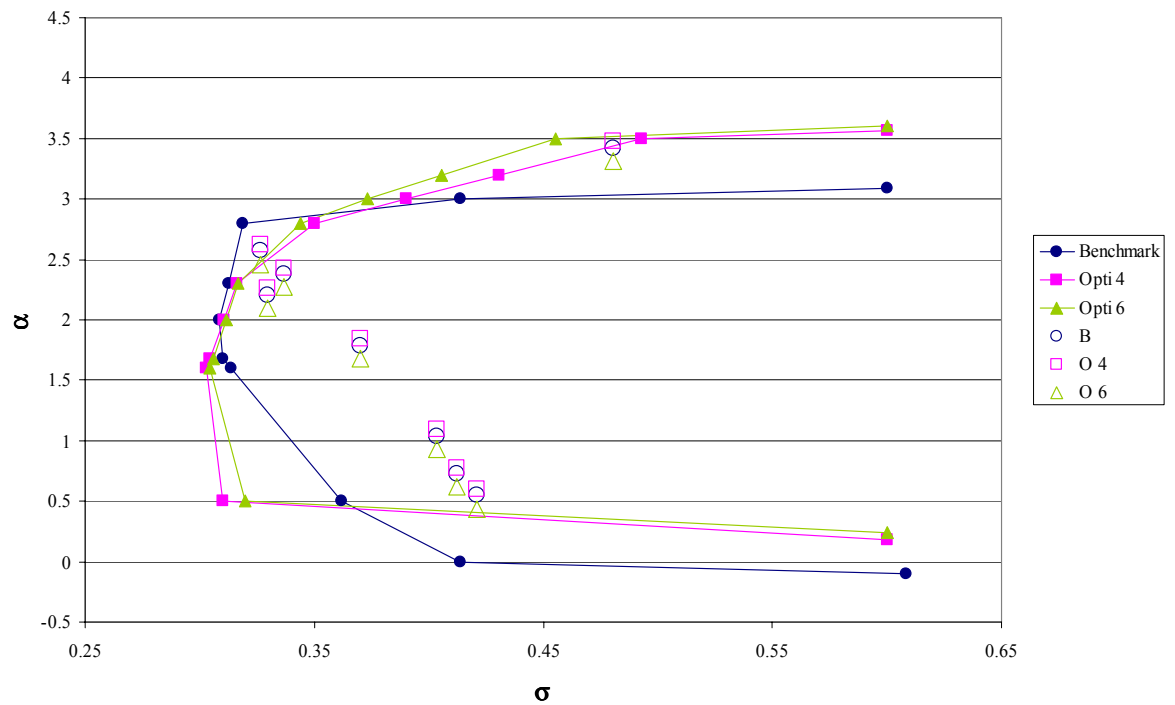


Fig.9: Comparison of performances of optimum solutions with 4 and 6 pairs of parameters

The most interesting feature of the proposed approach is that only some marginal parameters of input must be chosen by the user and these data are mainly related to numerical procedures used by the code, while hydrodynamics parameter, namely the arc limits and angles of attack, the beginning of pressure recovery region and of closure contribution in the upper and lower side of the profile are completely controlled by the GA. In this way also an inexperienced user can profit from the capability of computer programs such as Eppler's.

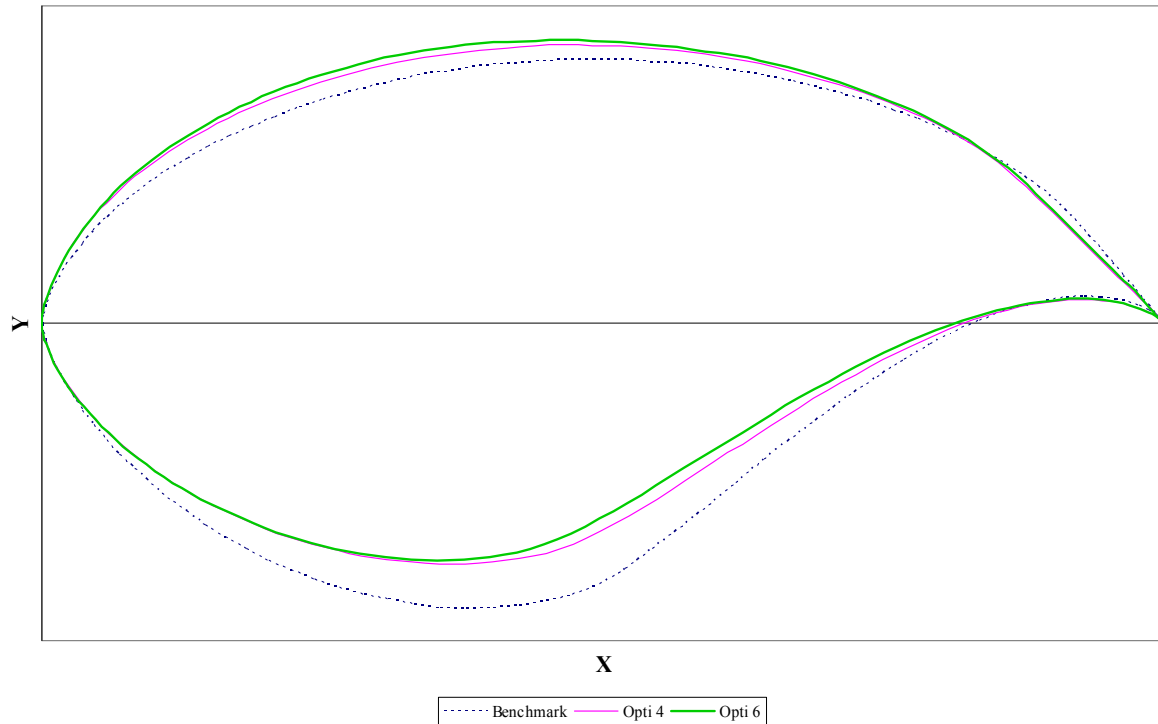


Fig.10: Section geometry of optimum solutions with 4 and 6 pairs of parameters

The GA let the designer to run several thousands of tests per hour also on a small PC and this assures a wide investigation over the possible solutions. Furthermore, the results can be conditioned by more complex fitness functions defined by the designer according to the particular needs of the problem. The consistency of the results is assured by the profile comparison that is performed taking into account the differences in zero lift angles.

References

- EPPLER, R.; SOMERS, D.M. (1980), *Supplement to: A computer program for the design and analysis of low-speed airfoils*, NASA TM 81862
- KUIPER, G.; JESSUP, S.D. (1993), *A propeller design method for unsteady conditions*, SNAME Trans. Vol. 101, pp.247-273
- KIRKPATRICK, S.; GELATT JR., C.D.; VECCHI, M.P. (1983), *Optimization by simulated annealing*, Science, 220, 4598, pp.671-680
- HOLLAND, J.H. (1975), *Adaptation in natural and artificial systems*, MIT Press
- GOLDBERG, D.E. (1989), *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley

A Systematic Study on Posing and Solving the Problem of Pareto Optimal Ship Routing

Jörn Hinnenthal, TU Berlin, Berlin/Germany, hinnenthal@ism.tu-berlin.de
Stefan Harries, FRIENDSHIP – Systems, Harries@FRIENDSHIP-Systems.com

Abstract

The paper proposes an optimization approach to selecting the most advantageous route on the basis of hydrodynamic simulations. An elaborated example is given for an intercontinental container service (CMS Hannover Express) between Europe (Le Havre) and North-America (New York). Different weather situations for the North Atlantic are taken into account. Time loss, fuel consumption, accelerations and slamming are minimized on the basis of several performance models – for instance response amplitude operators for the ship's motions. The routes in adverse weather conditions are established as perturbations to a parent route in calm weather which is assumed to be the concatenated great circles between way points. Utilizing a B-spline technique, the number of free variables for describing both the course and the velocity profile is kept low. For solving the multi-objective, non-linear and constrained optimization problem the commercial package mode-FRONTIER has been successfully applied. Investigations utilizing various objective functions serve to adopt a Multi-Objective Genetic Algorithm (MOGA) to automatically fill up a solution space of feasible routes for the support of the decision-making process.

1. Introduction

Sophisticated routing of ships is increasingly recognized as an important contribution to safe, reliable and economic ship operations. The more accurate both weather forecasts and performance simulations of ships in a seaway become, the better they serve to identify the best possible route.

An optimum route complies with the desired time of arrival at minimum fuel consumption and maximum safety. Permissible and reasonable loads on the ship, the cargo and the crew are not allowed to be exceeded. This establishes a multi-objective, non-linear and constrained optimization problem in which a suitable compromise is to be found between opposing targets.

2. Wave data

While the estimated time of arrival (ETA) and fuel consumption (FUEL) depend on all environmental conditions, accelerations and slamming are primarily influenced by swell. Swell - i.e., long-crested waves - being the crucial factor, we decided to focus on this influence, leaving out wind-sea, current, wind, drift-ice, shallow-water etc. for later and more refined work. The objective therefore was to suggest a novel approach to routing and to investigate its merits without claiming completeness of the model. It will be shown, however, that an extension of the optimization scheme is rather straightforward, allowing to quickly incorporate refined analyses.

The wave data used were provided by the *European Centre of Medium-Range Weather Forecast (ECMWF)*. They are a hind cast of the sea conditions in the North Atlantic in the periods from 6. to 16. of June 2001 (calm sea condition), 20. to 30. of January 2002 (rough sea condition) and 1. to 11. of January 2003 (intermediate sea condition) and are based on a reanalysis of buoy- and satellite measurements. The data for the significant wave height $H_{1/3}$, the peak period T_p and the wave angle β are given in time steps of 12 hours on a grid with 1.5° mesh size. Within the route optimization these data were treated as if they had been a wave forecast.

To obtain a continuous wave energy spectrum as a function of the wave frequency the data were associated with the description of a Bretschneider wave spectrum, cp. *Lewis (1998)*:

$$S_{\zeta}(\omega) = \frac{172.8 \cdot H_{\frac{1}{3}}^2}{T_1^4} \cdot \omega^{-5} \cdot \exp \left\{ \frac{-691.2}{T_1^4} \cdot \omega^{-4} \right\} \quad (1)$$

$\omega_p = \frac{2\pi}{T_p}$, the modal frequency, $T_1 = 0.772 \cdot T_p$, transformation from mean to peak period, *Journée (2001)*.

3. Ship responses in waves

Response amplitude operators (RAOs) and response functions for the added resistance due to waves (RFs) were employed for the evaluation of the ship performance in waves. Both have been calculated prior to the optimization by means of the strip theory code SEAWAY, *Journée (2001)*, based on potential theory. Besides RAOs for the center of gravity motions for all six degrees of freedom and RAOs for the motions of selected points also relative to the free surface, two different methods are available to determine the added resistance due to waves: The radiated energy method of Gerritsma and Beukelman and the integrated pressure method of Boese, *Journée (2001)*. The numerical results used within the optimization procedure are:

- The RAOs of the translatory motions on the bridge (calculation of the loading on the crew by means of accelerations).
- The RAOs of the motion relative to the free surface (calculation of the slamming probability).
- The RFs for the added resistance due to waves (preferably according to the integrated pressure method which displays a smoother distribution over the wave frequency ω than its radiated energy counterpart).

Fig.1 shows the RAOs of motion on the bridge, relative motion at the bow and the RFs of the added resistance due to waves as computed with SEAWAY. The plots serve to illustrate the input to the response simulations within the route optimization.

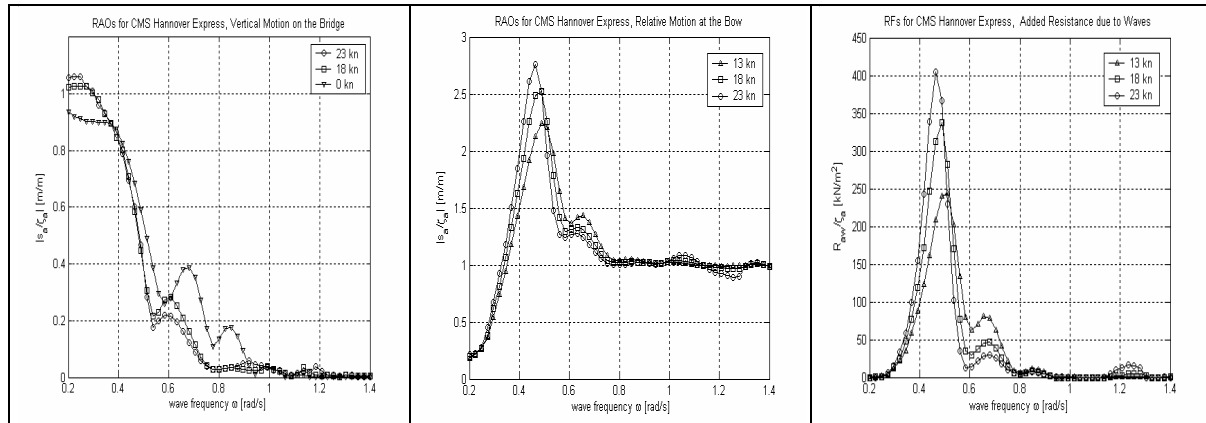


Fig.1: RAOs of motion on the bridge, relative motion at the bow and the RFs of the added resistance due to waves

3.1. Ship motions

The product of the wave spectrum S_z and a squared RAO of the considered motion yields the response spectrum of the ship motion S_s :

$$S_s(\omega) = |RAO|^2 \cdot S_{\zeta}(\omega) \quad (2)$$

Taking into account the ship velocity, this operation has to be performed with all involved terms

expressed as functions of the wave encounter frequency ω_e . Generally the n^{th} moment of the response spectrum is built with:

$$m_{ns} = \int_0^{\infty} S_s(\omega_e) \cdot \omega_e^n d\omega_e = \int_0^{\infty} S_s(\omega) \cdot \omega^n d\omega \quad (3)$$

The significant amplitude of the acceleration is determined with the following approximate formula provided by probability calculations:

$$\ddot{s}_{a/3} = 2.0 \cdot \sqrt{m_{4s}} \quad (4)$$

This can be applied to determine the acceleration on the bridge for instance.

3.2. Slamming

Slamming can be defined as the coincidence of two events: the emergence of the bow and its subsequent dunking above a critical velocity. The critical velocity according to Ochi, *Journée (2001)*, is:

$$\dot{s}_{cr} = 0.0928 \cdot \sqrt{g \cdot L_{pp}} \quad (5)$$

Presuming statistical independence of these events the probability of slamming is calculated from:

$$P\{slam\} = \exp \left\{ \frac{-D^2}{2 m_{0srel}} + \frac{-\dot{s}_{cr}^2}{2 m_{2srel}} \right\} \quad (6)$$

Where m_{0srel} and m_{2srel} are the 0^{th} and 2^{nd} moment, respectively, of the response spectrum for a point on the keel line 10% behind the forward perpendicular FP, D being the draught 10% behind FP.

3.3. Fuel consumption

The added resistance due to waves is calculated from:

$$R_{AW} = \int_0^{\infty} RF \cdot S_{\zeta}(\omega) d\omega \quad (7)$$

The determination of the calm water resistance is based on a regression analysis of model tests and full-scale data, following *Holtrop and Mennen (1984)*. The operating point of the propeller was calculated according to the *ITTC (1978)* power prediction method, using the characteristic of a propeller with the same diameter, number of blades, similar pitch and blade area ratio as given in *Yasaki (1962)*. In a final step the fuel consumption and the compliance with the permitted operating condition was determined in agreement with the project guide available from the producer of the ship's main engine, *MAN B&W (2000)*.

4. Setting up of the optimization task

Simply put, automated optimization is the formal process of finding a good (the best) solution from a set of feasible alternatives. It requires a complete mathematical problem formulation in terms of objective functions (what is to be improved), free variables (what shall be consciously changed) and constraints (what restricts the feasibility).

4.1. Free Variables

To provide free variables for both the spatial and the temporal description of the route the path and the velocity profile were given as B-splines. Perturbations of a parent route in time and space were realized by superposing the appropriate parent spline with a Greville-spaced shift spline. The vertices of the shift splines are controlled by shift parameters that were taken as the free variables of the optimization task.

The velocity perturbation was performed as a reduction of the design speed of 23kn. The spatial perturbation provides a shift of maximum 10% of the arc length of the original parent route to either starboard or port.

Fig.2 presents example routes as realized with 5 parameters for the spatial shift. The optimization was focused on the open water part of the journey, pilot time and estuary traveling was deliberately left out of the investigation. The geographical feasibility was checked during the optimization process.

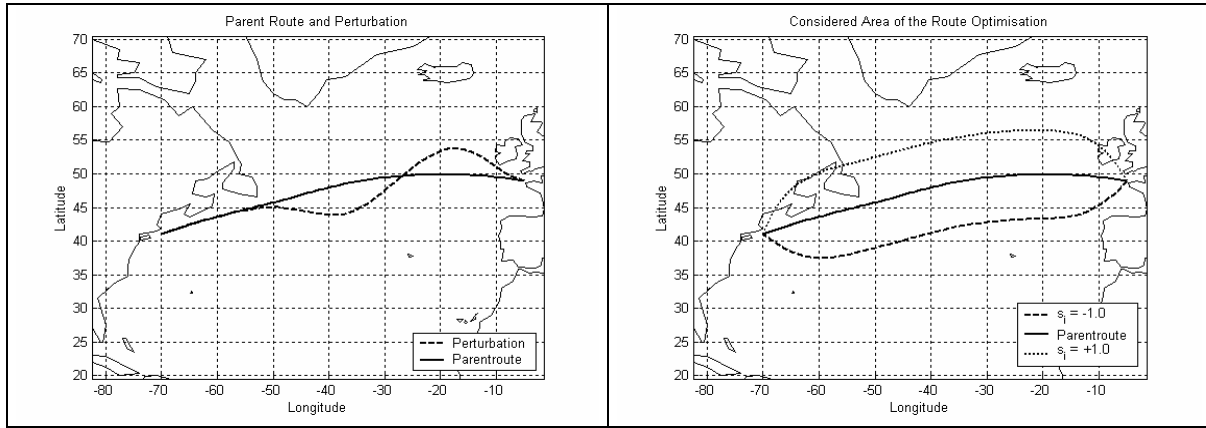


Fig.2: parent route, perturbation and investigated area

4.2. Constraints

Constraints were imposed by defining limiting values for the slamming probability (3%) and for the lateral and vertical accelerations (0.2 g , $g = 9.807 \text{ m/s}^2$).

4.3. Objectives

Two major objectives, the estimated time of arrival (ETA) and the fuel consumption (FUEL), were taken into account both of which had to be minimized. Especially in rough weather conditions initial route variants need to be quiet slow to avoid infeasibility due to active constraints. The following objective functions were used to accelerate the propagation of the initial routes towards faster route variants in the multi-objective optimization:

$$\text{Objective } ETA = (ETA - \text{set value } ETA)^2, \quad (8)$$

$$\text{Objective } FUEL = \text{weight} \cdot \left(FUEL - FUEL \cdot \frac{ETA}{\text{set value } ETA} \right)^2. \quad (9)$$

The *set value* of ETA was derived from optimum results at calm water conditions ($ETA = 114$ hours after departure, the value matches the estimated duration of a journey as stated in the sailing lists of the ship operator; the weight assigned to the FUEL objective was 0.05, so as to produce objectives of an equal magnitude at the biggest deviants of the set value).

4.4. Optimization with modeFRONTIER

For the optimization task the generic optimization software modeFRONTIER was utilized, www.esteco.it. The process flow in Fig.3 can be read from left to right. Initially, the free variables used as input parameter of the optimization loop are collected in an input file (parameter.txt). Afterwards an application is started to build the new route by means of a perturbation of the parent route according to the actual input parameter (makeroute). Next, the route is evaluated (virtualShip) and results are given back to modeFRONTIER. Finally, the optimization routine observes the constraints, calculates the objectives and, if necessary, launches a new investigation. The optimization results presented in the following chapters belong to a westbound Atlantic crossing of the Hapag Lloyd panmax container vessel “CMS Hannover Express”.

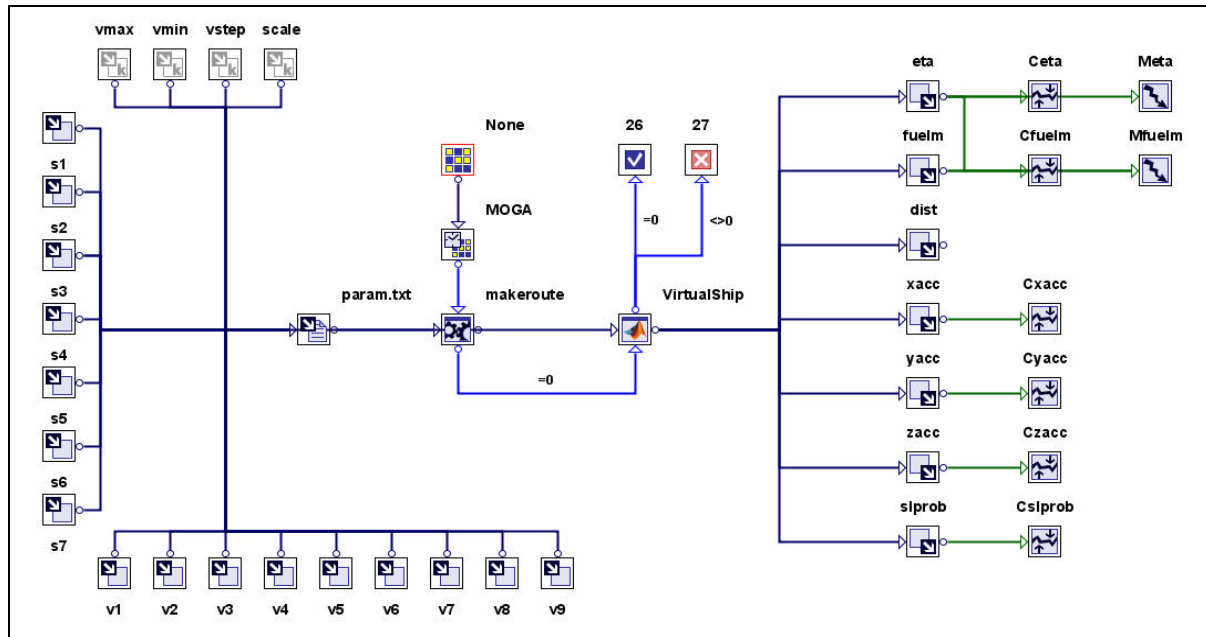


Fig.3: modeFRONTIER flow chart

5. Optimization results at different weather conditions

Two different search strategies, a SIMPLEX and a multi-objective genetic algorithm MOGA, were applied. Both, SIMPLEX and MOGA optimizations were conducted for a calm, a rough and an intermediate weather situation. Since the SIMPLEX algorithm can handle only one objective, optimizations were constricted to the minimization of ETA. Within MOGA the minimization of both ETA and FUEL was undertaken.

Fig.4 shows the result of a MOGA optimization in rough sea conditions. Each square represents a feasible route that can be taken into account for a route decision considering ETA and FUEL. The solution space is bordered by a Pareto frontier – the set of all solutions for which a single objective cannot be further improved without deteriorating any other objective. All routes lie above and to the right of the Pareto frontier. Those variants that are closest to the frontier display low passage time for reasonably low fuel consumption. Apparently, it is impossible to decrease ETA below certain limits without impairing FUEL.

Fig.5 illustrates the operation of a SIMPLEX optimization under rough sea condition. Starting from a set of initial designs at a low speed level the optimization converged to a local optimum. None of the routes touch the Pareto frontier found by the genetic algorithm. The figure nicely shows the limitation of a deterministic search strategy like the SIMPLEX algorithm. As a general rule it is difficult to identify a global optimum and the search goes quickly to the (nearest) local optimum. While neglecting FUEL an agreement of the optimization result with the *Pareto* frontier should therefore not

be expected. Further investigations with different starting points and objective functions depending on both ETA and FUEL achieved no improvement.

Table 1 and Fig.6 compare SIMPLEX and MOGA optimization results at different sea conditions. The time minimum routes of the SIMPLEX and *Pareto* routes of the MOGA are depicted. For the genetic algorithm 7000 route variants were investigated (one population consists of 70 individuals, 100 generations). The SIMPLEX converged most of the time with less than 400 iterations but never matched the time minimum route found by MOGA. An advantage of the SIMPLEX is time saving due to smaller computational effort. The MOGA, however, is capable to fill up the solution space, to identify a Pareto frontier and therefore to provide a meaningful data base for the routing decision support. Each route presented by a square in Fig.4 can be selected and taken into account for the decision making of the naval officer considering ETA and FUEL. But also individual preferences concerning slamming probability or allowable accelerations can be regarded. Consequently a more conscious decision can be made about which route to take. Sometimes this will yield fuel savings for arriving at the desired time. Sometimes this will show which route is still the best (or least worse) for severe conditions. Furthermore, only the MOGA found the shortest route at calm sea condition (although maximum significant wave heights of about 5 meters occurred during the crossing, the constraints for slamming and accelerations did not affect the optimization as they all turned out to be inactive, therefore the time minimum route has to be the shortest one).

Fig.6 nicely shows the influence of the sea condition to the minimum attainable ETA. In rough weather an arrival on schedule was impossible. The different fuel consumptions were caused by the ship speed, the necessary distance of the journey to avoid severe weather conditions and also the added resistance due to waves.

Fig.7 illustrates four steps of an example route in rough weather, here with lowest ETA (143h). The contour lines represent the current significant wave height during the crossing, the route and the actual position of the ship (a dot at the end of the bold black line) are drawn, too. The decrease of fuel consumption in comparison to the optimum route at calm sea condition is caused by low ship speeds during the first 50 hours (below 18kn). From an optimization point of view this was necessary to let a wave field, crossing the route from south to north, pass (first picture of Fig.7). The fuel savings due to the reduced velocity exceeded the additional fuel consumption due to waves and due to a longer distance of the journey.

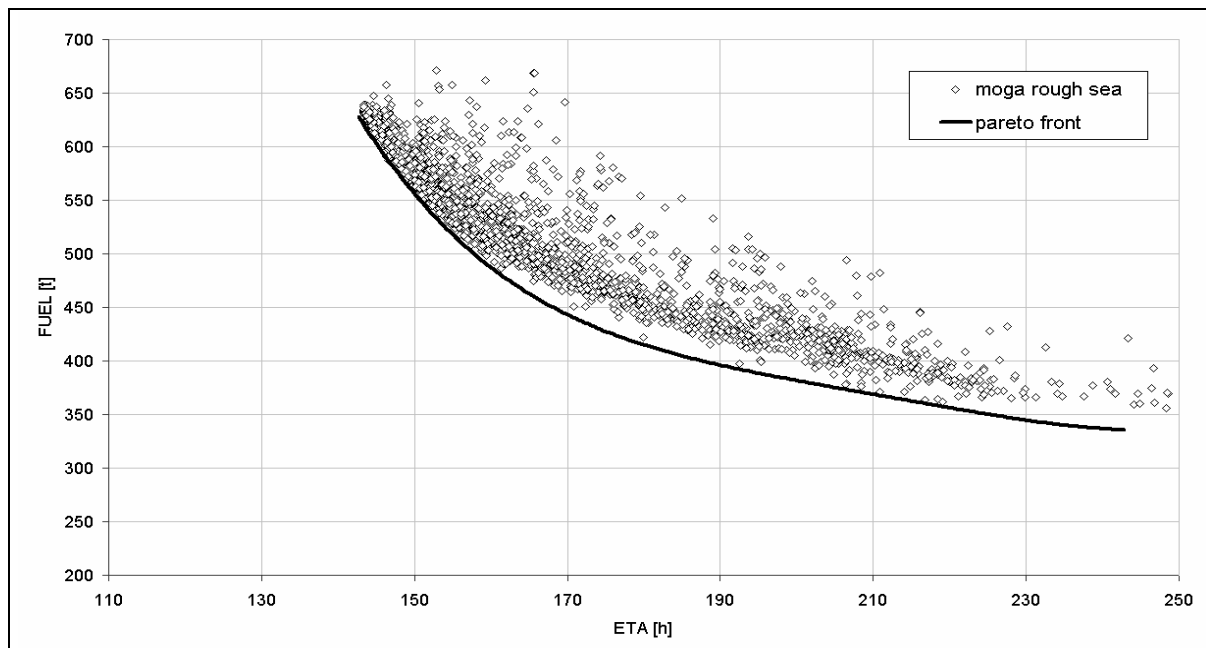


Fig.4: MOGA results and Pareto frontier

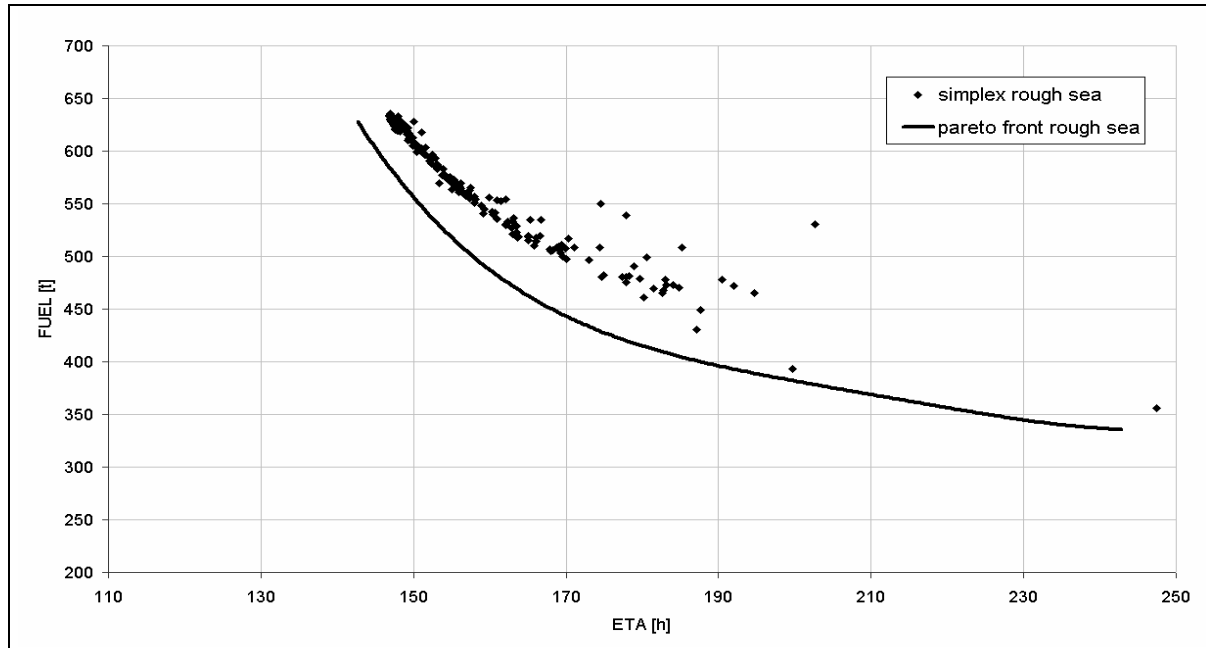


Fig.5: SIMPLEX optimization and Pareto frontier obtained by MOGA

Table 1: time minimum routes for different optimization strategies and sea conditions

ETA distance	FUEL design number	MOGA		SIMPLEX	
calm sea condition		114 h	696 t	118 h	624 t
		2716 nm	7000	2722 nm	392
intermediate sea condition		119 h	686 t	121 h	713 t
		2717 nm	7000	2789 nm	302
rough sea condition		143 h	639 t	147 h	633 t
		2756 nm	7000	2813 nm	368

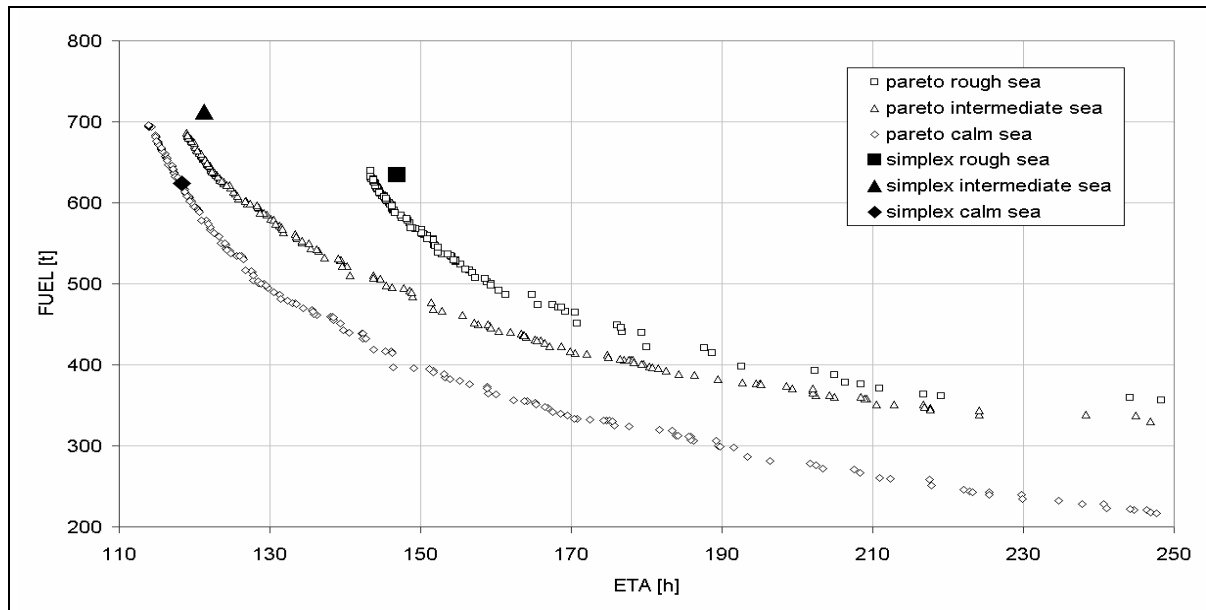


Fig.6: MOGA Pareto designs and time minimum routes obtained by SIMPLEX

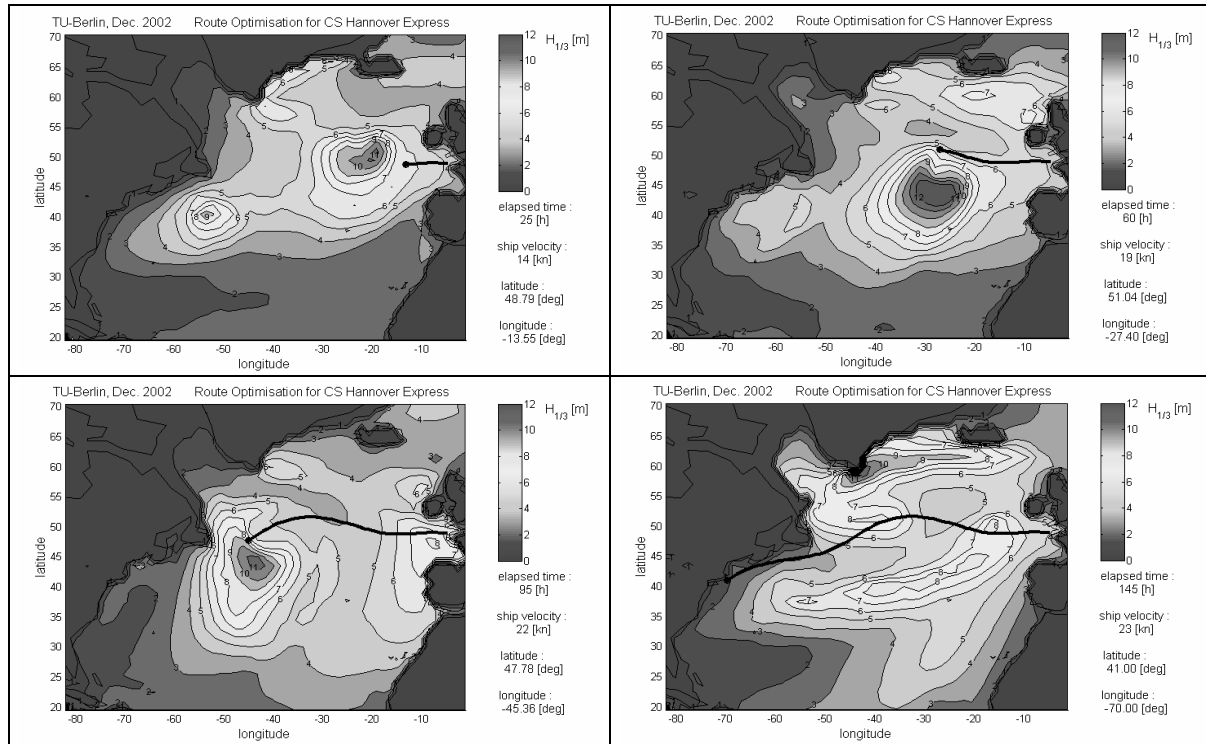


Fig.7: Optimized route, ETA = 143 h, fuel consumption 639 t

6. Summary and conclusions

Initially the paper provides a brief introduction into the mathematical description of a ship in waves used within the presented route optimization. A new approach for the spatial and temporal generation of design variants based on parametric curves has been presented. Investigations concerning the number of free variables showed that at least 9 variables for the description of the velocity profile and 7 variables to describe the course are necessary to obtain a sufficient temporal and spatial resolution of the route that correlates to the wave pattern as well as to the geographical conditions in the North Atlantic. The chosen B-spline description turned out to be a reliable method for the set-up of the route optimization task. The major advantage of parametrically modeled route- and velocity profiles lies in the reduced number of free variables. This enables the application of a stochastic algorithm with a reasonable time exposure.

Comparative investigations with different optimization methods showed that the algorithm must be able to avoid a premature stop at local optima. Two strategies, the deterministic SIMPLEX and the stochastic MOGA were utilized here. With the MOGA one may identify meaningful optima. (Other strategies, for instance simulated annealing, might also be considered but were beyond the scope of this study.) The decision making of the naval officer is supported by building up a significant amount of information from which the best compromise (Pareto optimum) can be found in accordance with individual preferences. A further aspect that should be kept in mind in the future is the probabilistic character of the weather and its forecast, *Hoffschmidt et al. (1999)*. As the optimized routes are guided closely along areas of rough sea condition, Fig.7, the reliability of the forecast has to be included into the evaluation of the optimization results and robustness ought to be taken into account.

Acknowledgement

The work presented here was substantially facilitated and improved by Oyvind Saetra of the European Centre of Medium Range Weather Forecast, Reading, UK, who provided wind and wave data. Furthermore, some of the work reported here was supported by the European Project SEAROUTES, G3RD-CT-2000-00309.

References

HOFFSCHILDT, M.; BIDLOT, J-R.; HANSEN, B.; JANSSEN, P.A.E.M. (1999), *Potential benefit of ensemble forecast for ship routing*, ECMWF Techn. Memo. 287, Internal Report, ECMWF, Reading/UK

HOLTROP, J.; MENNEN, G.G.J. (1984), *An approximate power prediction method*, International Shipbuilding Progress, Vol. 31

ITTC (1978), *Performance prediction method for single screw ships*, 15th Int. Towing Tank Conf., The Hague

JOURNÉE, J.M.J. (2001), *User and Theoretical Manual of SEAWAY, Rel. 4.19*, TU Delft

MAN B&W (2000), *K90MC MK6 Project Guide Two-stroke Engines*, 5th Edition, www.manbw.dk

LEWIS, E.V. (Ed.) (1998), *Principles of Naval Architecture - Motions in Waves and Controllability*, SNAME, Jersey City

YASAKI, A. (1962), *Design diagrams for modern four, five, six and seven-bladed propellers developed in Japan*, 4th Naval Hydrodynamics Symp.

Concurrent Engineering through the Supply Chain from Ship Owner to Supplier

Otto Millbourn, Tribon Solutions, Malmö/Sweden, otto.millbourn@tribon.com

Abstract

Efficient shipyards of today use a 3D Product Information Model as the main backbone for their design, planning and production. In 2000, TRIBON Solutions AB released Tribon.com, an Internet based, searchable shipbuilding component database. The database contains searchable and structured information and 3D models of shipbuilding equipment. Tribon.com brings the supplier closer to the design process of the yard and makes collaborative engineering possible. The idea to complete the supply chain and include the ship owners comes naturally. Today the communication during the new building process is often cumbersome and costly. The most expensive option, on site inspections, is used to a great extent. The new process offers an integration of the Product Information Model and Tribon.com. It enables the ship owner to monitor the design and progress of the new building via the Internet. The ship owner will be able to comment on the design, follow up milestones, simulate operations, and get sufficient information on equipment - in real time. Some ship owners and yards already test the new process. The benefits for all stakeholders are substantial: less rework, fewer man-hours spent, fewer misunderstandings, increased onboard safety and others. The next step is for the ship owner to use the Product Information Model and Tribon.com to decrease the life-cycle cost.

1. Introduction

Today shipyards and ship owners communicate in different ways in the new building process (usually after the order is signed). The communication is extensive and relatively unstructured. The yards communicate with classification societies, suppliers and owners. The methods are fax, e-mail and ordinary "snail mail". In most cases, the documentation is sent as paper-based 2d drawings and specifications. Due to old traditions and the competitive situation among various suppliers, the individual supplier tends to reveal its technical and commercial "features" at "as-late-as-possible" stage. In addition, the owner communicates internally between staff on the building site and the head office.

The communication and all the papers/documents are a problem for both the yards and the owners because of misunderstandings in 3D to 2D conversions and the extent of the information. This problem has been expressed by a number of yards and owners: Could we give access to the PIM via Internet, could Tribon.com be one information channel? The current process is costly and cumbersome.

2. Tribon components

2.1. Digital Product Model

A product model can be regarded as a comprehensive "Ship Database" containing all relevant information about a specific ship. The Tribon Product Information Model is object-oriented in the sense that all design and production data are stored and handled as objects. These objects represent all of the types of physical items found in shipbuilding (systems, components, assemblies, pipes, equipment, cables, plates, stiffeners etc.). The description of each object type is formulated such as to contain all the necessary technical data and/or properties required for the description of a particular instance of the object. The technical data and properties are then used to derive the graphical representation of the object for use in symbolic sketches or in 2D or 3D views, depending on the presentation context. When design modifications become necessary this technical data is changed rather than the graphical information, which would be the case in most general purpose CAD systems.

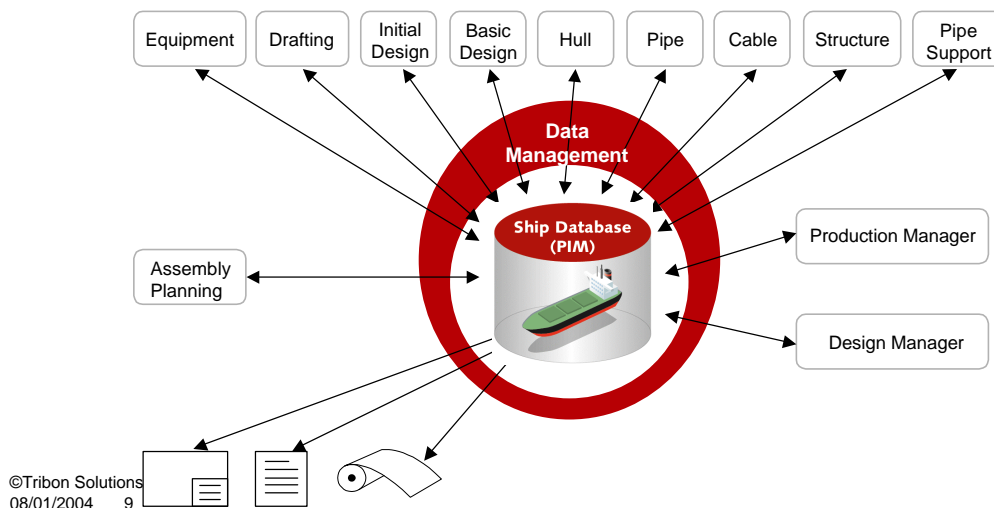
2.2. Tribon Product Information Model

The Tribon Product Information Model (PIM) (and the related software, the Tribon System) plays the role of the central tool for design, engineering and material requirements definition in the shipyards. Since all product-related data is stored in one place (the PIM) it also works as the “hub” for planning and coordination of the ships arrangements from a macro-perspective as well as on a detailed level. When working with traditional methods of drawings (paper drawings or electronically stored documents) there is no real mechanism to coordinate the development of the arrangements simply because many of the designers work in parallel in a highly concurrent engineering environment. Different systems in the same room (compartment) are produced in different drawings, which creates risks of collisions in the final (physical) arrangement. Each avoided collision represents a substantial saving in terms of a lower degree of rework during the assembly process. One could claim that “PIM produces better ships”. (Reports from shipyards using the Tribon PIM technology show 30% savings in design and engineering man hours and 8% savings in production man hours, compared to previous use of systems based on 2D electronic drawings philosophy.)



The PIM

- A valuable asset for maintenance. Equipment information, information for repair work, naval information ...

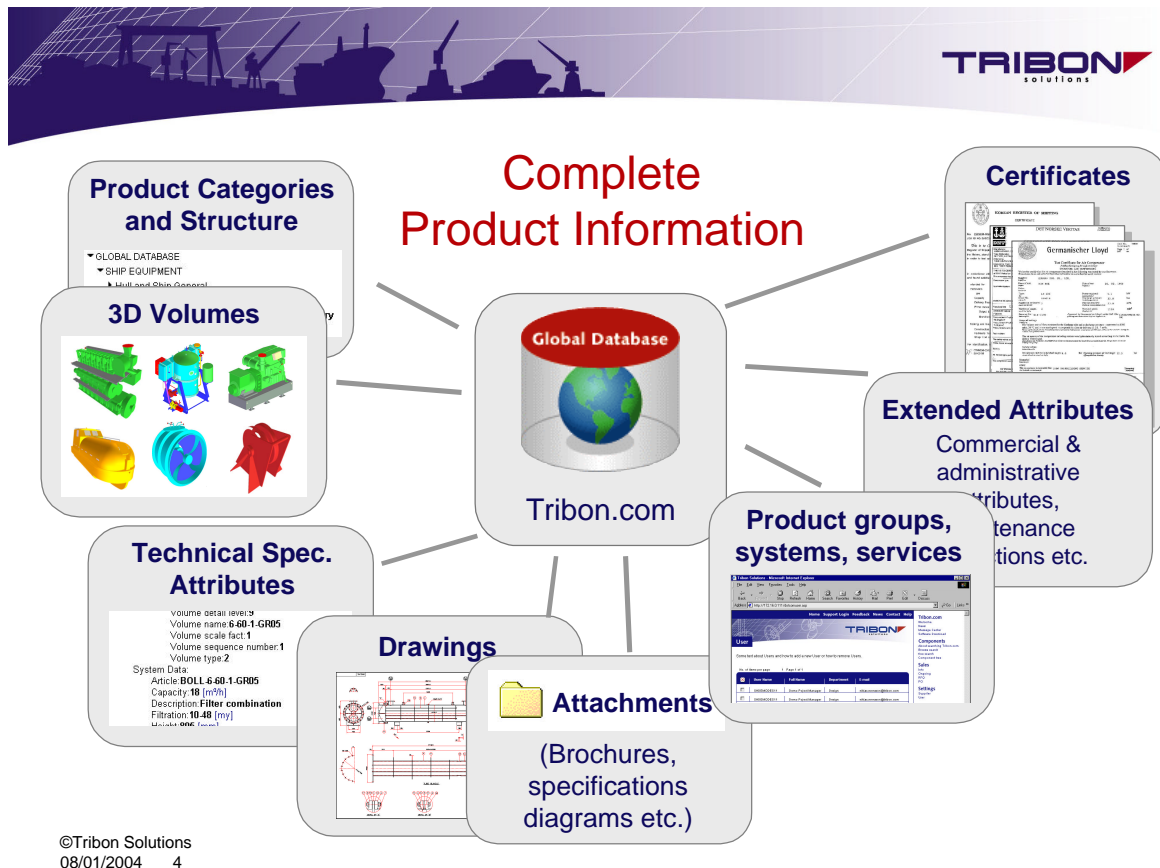


2.3. Tribon.com

Tribon.com is a neutral Internet-based service for the maritime industry. It links shipyards and marine equipment and system suppliers in a comprehensive and unique global network. The service revolves around a global database that contains technical information and components and equipment. By subscribing to the service, shipyards can access, download, and integrate information directly into their design model. The instant access to accurate, complete, and standardised product information will improve and shorten the time spent searching for component information. Designers are able to reduce lead times and have global access to supplier products. This is the core functionality of Tribon.com.

Essentials:

- A neutral Internet-based service that gives instant access to accurate, technical and updated information in a standard shipbuilding format to suppliers, shipyards and design agents
- Enables easy design integration
- Contains data in standard formats
- Is complemented by e-commerce functionality
- Access information 24 hours a day, 365 days a year
-

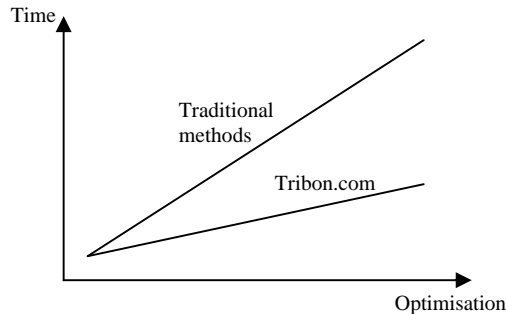


Suppliers know how time consuming and costly it is to handle information requests from customers and to get their components or systems specified to suit the needs of shipyards repeatedly. Tribon.com is a flexible marketing channel that allows suppliers to be in control of their information and update it as their products or systems develop. Through Tribon.com suppliers can also broadcast their information and news to designers. Tribon.com provides many possibilities for suppliers:

- Access an efficient marketing and sales process
- Reach engineers and designers at shipyards in the global shipbuilding market, your target group
- Create the information once - share it with their customers concurrently, refine it and use it many times
- Improve communication, quality of data and decrease the paper trail
- Get a new channel to the after-sales market
- Increase customer satisfaction

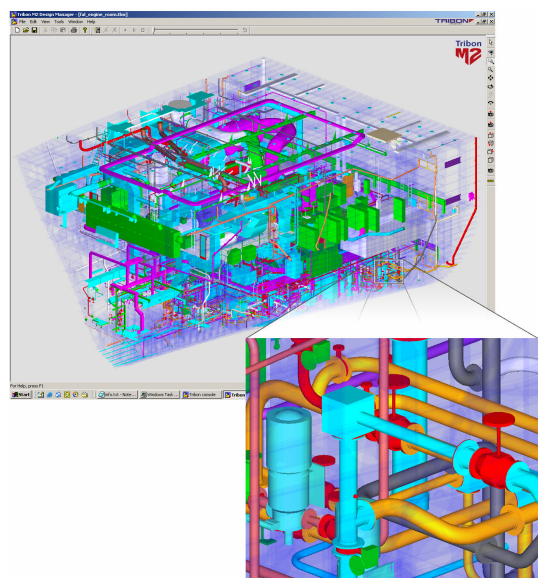
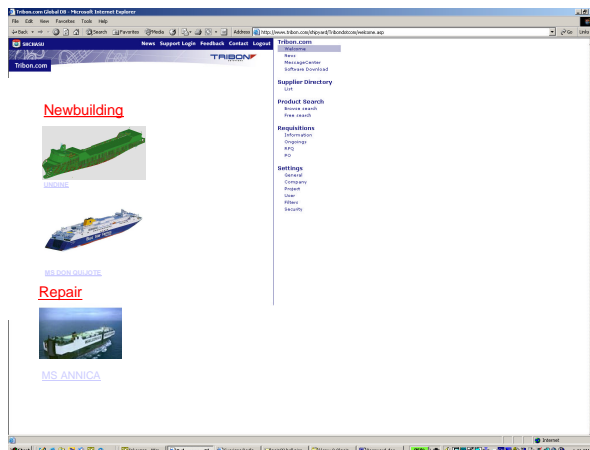
3. New Process

The new process is being tested with several ship owners: BP, Wallenius, TOTE, and the US Navy. Since the functionality is not yet developed, we have simulated the work process. The communication is made via the Internet, but not yet inside Tribon.com which has to be driven side by side. The tests have concluded in the simplified process below. Also the benefits described are concluded with the help of the tests. The results from the tests were so beneficial that some of the owners have continued with the simplified solution and are now using it in operation.



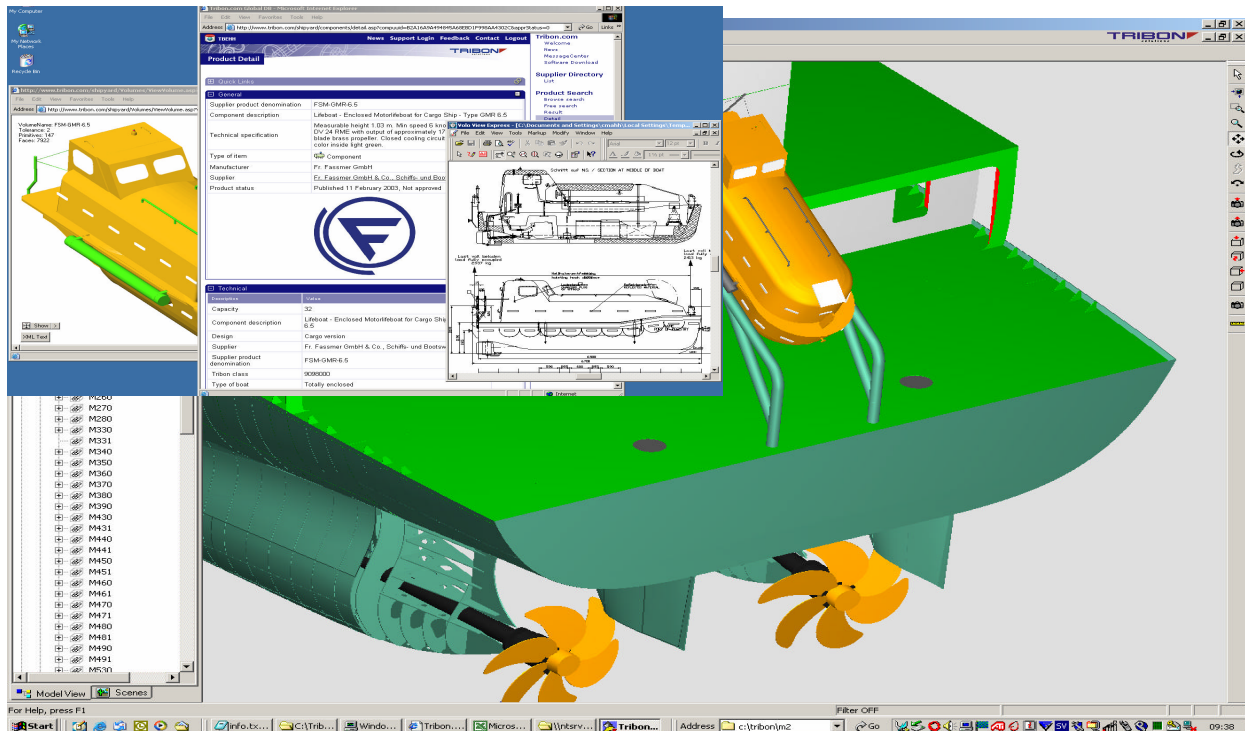
All engineering strives for to the “optimal design”. Often time and economy stop the engineer from reaching total satisfaction, especially in commercial shipbuilding.

Research has shown that 30% of an engineer’s time is spent on “searching for information”. Tribon.com gives the engineer direct access to the information, unlike traditional methods where he would spend too much time to search.



The new process builds on the opportunities that the Internet gives. Ship owners log in via Tribon.com and can call up their new building or repair project. After choosing project, ship owner can browse around in the 3D model and:

- Check and comment the design in the model
- Perform analysis, test functions (allowing staff to “learn” the ship before it is built)
- Follow up the progress of the project. By clicking different objects the user can get status on the objects. Example of status could be:
 - 1) for designs: in design, waiting for class approval
 - 2) for equipment: supplier set, procured, on stock, assembled.
 - 3) For material produced by the yard: design approved, in production, assembled
- Port staff and suppliers of port cargo equipment can use the model to test port equipment
- The user is able to click equipment and if the supplier is set the user can get detailed information of the equipment. Tribon.com contains links to supplier websites and e-mail addresses; so the owner can connect directly for questions and discussion.



3. Benefits and risks

Benefits for the owner in a new building or repair project are:

1. Instead of calling for information from the yard they can pick up the information themselves. Not having to send drawings and other information will save man-hours and cost.
2. This tool can improve communication with the main office and the onsite office reducing travel cost.
3. The possibility to access the design during the basic design stage will allow the owner better to affect the design, so that right decisions are made from the beginning and costly rework could be avoided.
4. Crew and other users can get familiar with the ship via Tribon, reducing rework and training cost.
5. Port staff and port equipment suppliers can use the design to test port equipment.

Benefits for the yard in a new building or repair project are:

1. Instead of sending information on demand, the yard gives access to the information, and the owners can pick up the information themselves. Not having to send drawings and other information will save man-hours and cost.
2. Misunderstandings and disputes can be avoided, since the yard always has access to the design.

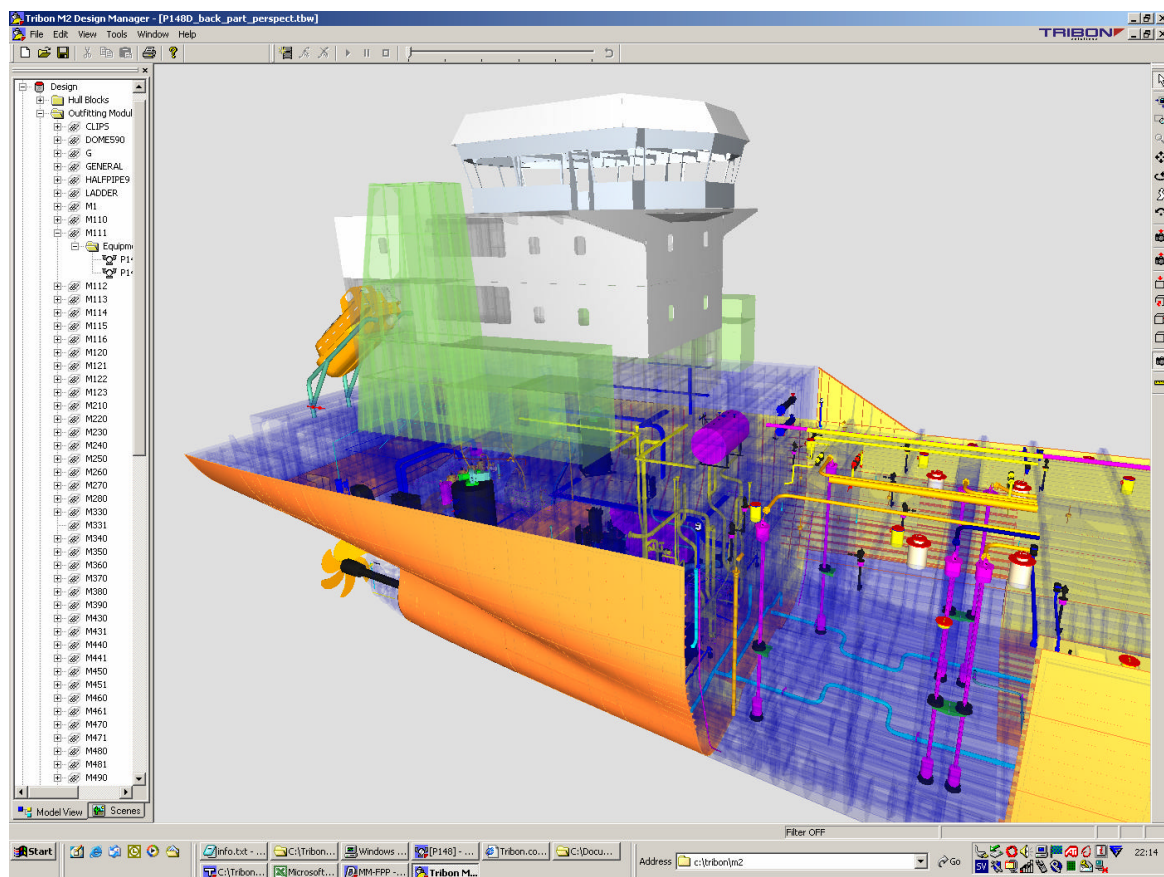
Some yards could feel that it is risky to put up the PIM on the Internet, allowing intruders to get access to the network via the PIM and the Internet. In order to avoid this, an automatically created copy could be put on the Internet once a day. The copy could be put on a stand-alone PC with no access to the network.

4. Conclusions and next step

The new process is a step further in the usage of the PIM. The PIM contains information necessary for everybody involved in the process of building a ship. With Tribon.com, Tribon allows now to distribute the already created information to those who need it, benefiting all parties involved with substantial cost savings and boosting the overall process. To emphasise the conclusion, I quote Mr. John Boylston, vice-president of new constructions at the ship owner Totem Ocean Trailer Express (TOTE), one of the early adaptors, from an article in the American ship review 2002-2003:

“Therefore, taken to an optimistic extreme, little or no rework should be necessary in construction. A rule of thumb is that a dollar spent in construction is equivalent to 10 dollars when rework is required. While we push the yards to become more efficient and to reduce construction cost, this author believes there are more significant returns in taking the required time to design in 3D and involving the owner in the process.”

The natural next step is for the owner to also get access to the PIM after the ship is delivered, to use the important maintenance information in the model, to have the PIM if repair work or conversions are necessary. To use the PIM for support of the life-cycle of the ship will further cut cost for everybody involved in the shipbuilding business.



An E-Commerce Platform for Engineering Services in Shipbuilding

Hans Wuttke, Potsdam Ship Model Basin (SVA),
Potsdam/Germany, wut@sva-potsdam.de

Andreas Schwill, Institute for Computer Science, University of Potsdam,
Potsdam/Germany, schwill@cs.uni-potsdam.de

Frank Domeyer, Institute for Computer Science, University of Potsdam,
Potsdam/Germany, domeyer@cs.uni-potsdam.de

Abstract

We present intermediate results of a long-term project to develop an Internet platform that offers engineering assistance in naval hydrodynamics. Customers are able to use, for instance, prediction tools online, get quotations for standard orders and may look-up the progress of larger projects. These services on demand are a natural extension of e-business. But while trading goods over the Internet is almost standard, offering engineering services shows some specifics which by now are not fully understood. They are mostly unique and consist of multiple parts, require more information exchange and the corresponding order specifications are more comprehensive. This leads to problems that require new approaches and solutions in areas like application architecture, application development and integration, security and IT operations. This paper describes vision, development and implementation of the platform as well as experiences of customers.

1. Introduction

In general a shipbuilding project involves a large number of engineering services requiring by 15-20% of the total cost not mentioning – in particular for international contracts – the considerable costs for transactions and negotiations concerning order specification, transfer and return of ship data, discussions of results etc. Often these costs multiply since engineering services are ordered repeatedly with small changes and variations of requirements, ship plans, framework with respect to size and speed of the ship such that the overall contract spans several months.

Reducing these costs is a major objective for all companies and institutions participating in a project. A promising strategy is to transfer certain business processes on the Internet and to reduce the time needed to perform certain services eventually aiming at a turn-around time of 24 hours for selected standard services.

Table I: Characteristics of e-commerce systems for products and services

product-oriented e-commerce systems	e-commerce systems for engineering services
ad hoc ordering	intensive exchange of information precedes ordering
simple description and presentation of orders	description of orders by large data collections
short-term contact between seller and buyer, often anonymously („one-click shopping“)	long-term cooperation of business partners to perform services
different and comfortable payment functions	payment functions are less important
elementary products with few attributes	structured services consisting of many different subtasks
standardized mass-produced goods	almost unique services („one customer, one product, one process“)
stable quality guaranteed by brand (independent of seller)	varying quality (depending on service provider)
dominating paradigm: shopping cart	no paradigm known

2. E-Commerce of goods versus services

There are different e-commerce systems both for B2B- and for B2C-relations, e.g. for trading hard- and software, books, CDs, second-hand machines, in the wood market and many more. Moreover, tailored applications for customer relationship management, supply chain management and business intelligence are available. However, all these systems only support electronic trading of physical objects that are mostly branded goods, world-wide known and available anytime and anywhere at equal conditions and quality.

Only very few experiences are available for electronic sales and marketing of the product „service“ in particular of a unique service as it is in the case of engineering services. Available e-commerce solutions are often restricted to calls for tender based on simple questionnaires. Table I shows typical characteristics of e-commerce systems for products versus services.

The problems to be solved are manifold. For shipbuilding industry among the problems to be managed are the following:

- large data collections that describe the object of investigation
- different data formats constrain easy data exchange
- coupling of different IT-systems of contract partners
- iterative communication procedures towards the final product.

However, considering the development towards a service society in the 21st century it is essential to process complex services by electronic media in order to profit from cost reductions.

3. Vision of the E-Commerce Platform for Engineering Services in Shipbuilding (ePING)

The research and development in the field of ship hydrodynamics is one major activity of ship model basins. Their knowledge and experience is hidden in countless procedures, tools and technologies. The vision underlying our present efforts is to offer parts of this know-how via Internet in order to improve customer support. The added value of purchasing technology and know-how as a service is becoming increasingly interesting to the shipbuilding industry. Buying engineering knowledge as a service would allow many small and medium-sized enterprises to obtain best-of-breed integration solutions faster and cheaper, avoiding usual disruptions and expenses caused by continuous upgrades of software. Moreover, engineering services on demand can reduce the complexities and headaches associated with integrating new technologies.

The project ePING is a joint work of Potsdam Ship Model Basin (SVA), a non-profit company, and University of Potsdam and aims at conceptual development, feasibility, design and implementation of an e-commerce platform (B2B) for project-related communication, ordering, processing and billing of engineering services for shipbuilding.

The following typical scenarios illustrate main visionary features of the system as well as possible cost reductions.

Scenario 1: Public area with anonymous access.

By publications in a journal or on the Internet a shipowner got aware of SVA. In order to obtain an overview of SVA's services the shipowner uses the e-commerce platform – anonymously at first. He is able to use certain simple services with small input and output data sets. He enters the specification of the task and the data into a service form and sends it unencrypted to the SVA server. The system processes the job immediately using available technical algorithms (numerical procedures, CFD etc.) and returns results to the customer by email or displays them in a browser. Billing may be possible on a pay-per-use basis. Erroneous or incomplete as well as non-standard jobs are recognized, if possible, and rejected with an error message.

Scenario 2: Closed area for registered users with password protection.

A shipyard and SVA have established a long-term business relation. The shipyard is registered in the e-commerce system and uses its services on a regular basis. The customer, still in a planning stage of a new ship, wishes to use the engineering competence of SVA for a standard problem in ship design (e.g. wave resistance). He is also interested in a competent report in order to be able to quickly compare different variants of the ship with respect to cost-benefit ratio. The customer uploads relevant data to the SVA server and at the same time specifies his task as precisely as possible using a questionnaire. Data transfer is encrypted. The system classifies the task, checks data and task specification for completeness, correctness and consistency, preprocesses the task (decryption, data conversion and completion), assigns it to an SVA engineer as supervisor and starts – as far as possible – corresponding technical procedures. After a final check by the engineer the results are e-mailed to the customer. The customer may request a status report during all processing stages. Faulty, incomplete or non-standard tasks are recognized, if possible, sorted out and fed into the traditional working routines. Billing may be done on a pay-per-use or fixed-rate basis.

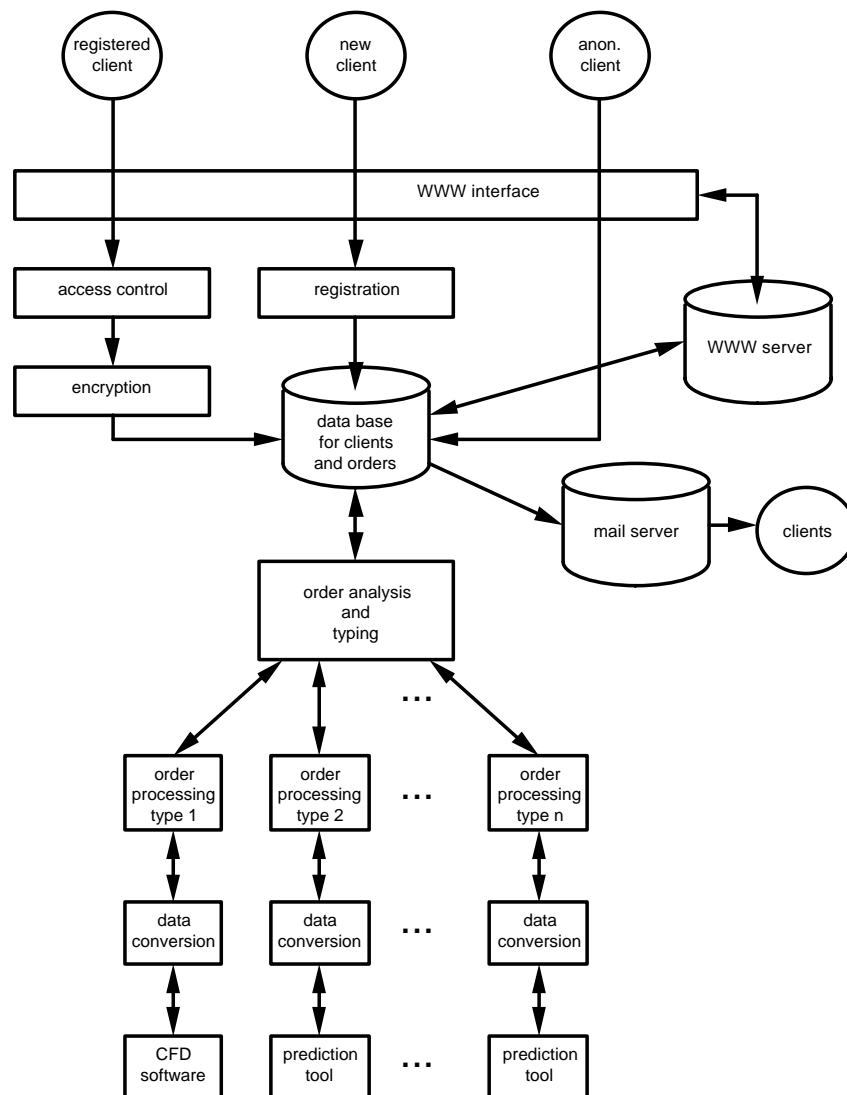


Fig.1: Architecture of an online platform for engineering services

The project is prototypical in the sense that considerations on system architectures, structures and techniques may be carried over in part to other engineering services, e.g. in architecture, structural analysis of buildings, test engineering, plant construction.

Main steps on ePING's road-map are:

- definition of a framework for processing engineering services, analysis of requirements in cooperation with pilot customers
- modeling by analysis and classification of typical business and service processes
- identification and definition of services and processes that may be standardized, structuring of these services based on few elementary service components that may be combined to complex services
- definition of a framework in which these basic building blocks may be further developed to tradable (branded) products of well-defined quality and functionality
- development of a concept for an e-commerce platform for project-related communication, ordering, processing and billing of services, research on suitable data exchange formats, protocols and security standards
- development, implementation and testing of the platform starting with simple and free services for a group of users among the customers of SVA; development of interfaces and conversion modules for the software systems (numerical algorithms, prediction tools, CFD) used by SVA
- transfer of main business and service processes to the e-commerce platform, evaluation and further development of proposed standards.

Fig. 1 shows the architecture of such a system.

3.1. Online ordering of engineering services in shipbuilding: A drawback

An essential issue in Fig. 1 is the module for task analysis, classification and typing. The corresponding problems in the fields of pattern matching, text analysis, data mining, knowledge presentation and acquisition etc. are subject of current research in computer science and widely unsolved yet. An overview is given in *Domeyer (2002)*.

Fortunately, in this particular case of engineering services for shipbuilding order classification and typing are not so essential since the number of orders per year is relatively small and most orders to SVA are neither anonymously given nor completely specifiable in advance nor to be billed electronically. But on the contrary before ordering most customers directly get in touch with the appropriate SVA department and try to communicate to an SVA expert on their specific tasks or inquiries. So when these person-to-person discussions eventually lead to a formal order, the task is already classified and assigned to the appropriate engineers. Table II lists sources and channels of orders to SVA in recent years.

Table II: Sources for orders to SVA (2001-2002)

regular customers by long-lasting business relations	40 %
customer visits (genuine order acquisition)	10 %
Recommendations	10 %
follow-up orders	20 %
new customers via Internet inquiries	3 %
new customers via presentation on fairs	2 %
new customers via lectures or publications	5 %
unknown, miscellaneous	10 %

A second issue which makes it very difficult and also obsolete to classify orders automatically is that in recent years orders have become more and more unique and non-standard. These tasks need preliminary development of models, concepts and technologies which may or may not end up in official orders. This long process of negotiations can hardly be supported online.

And thirdly, in particular in an early stage of product development, a customer wishes to confidentially talk to a competent partner eventually ending up in a joint concept to realize the task. This customer is not helped by a standard web form or a computer-generated response.

It follows that currently complex engineering services in shipbuilding neither may be sold over the Internet in a customer-friendly way nor would be much appreciated by customers due to extensive person-to-person communication needs.

3.2. Engineering assistance and value-added services over the Internet

Considerations in the previous section have encouraged us not to attempt to replace traditional communication, ordering and processing but to focus on assisting both customer and SVA engineer in typical problems and information and service needs which may be easily performed over the Internet. Consequently, the objective has to be to improve service processes in order to generate added-value and to satisfy customers.

Which service and information processes in shipbuilding generate added-value when transferred to the Internet? By analyzing SVA's business processes as well as customer surveys on using the Internet as an additional communication platform we came to the following conclusions:

The processes of ship model basins are based on deep customer relations lasting for several months, or even years if follow-up projects are involved, and are dominated by extensive communication between customer and SVA engineer where access to two categories of services is frequently required by customers: *Information on demand* and *computation on demand*.

Information on demand covers all information that describes the framework, environmental conditions, concerned engineers and development stage of the project. Distribution of this information over the net is very efficient and quick access to all relevant data gives customers an added-value and reduces communication with SVA engineers.

Computation on demand concerns services which offer online procedures for complex calculations in ship design. There are at least two advantages: Firstly, the software is available for different computer platforms with small requirements on resources (standard browser and Internet access) while there is only one copy to be maintained (ASP=application service providing). Secondly, part of the know-how of SVA which is hidden in countless small and medium-sized single-user programs covering specific problems on ship design, often written from scratch by an SVA engineer and mostly roughly documented not only turns into a marketable asset but also becomes fully available on the intranet. While employees of the company gain full access to all tools in-house, external users, of course, only have partial access.

These two application fields have been selected to be covered by our platform ePING in the first stage and will promise a considerable additional benefit for ship model basins and their customers.

4. Realization

Our revised system concept, Fig.2, includes the considerations above and experiences and is oriented towards SVA's corporate strategy. The collection of various procedures, methods and communication possibilities form the platform's backbone and is available via the SVA homepage <http://www.sva-potsdam.de>.

One of the trends in e-commerce is personalized access to information, *Finkelstein (1999)*, where customers gain access to specific information relevant for their purposes which may be realized by different user profiles that allow different views onto SVA's business activities. This paradigm of individual communication and services based on user profiles guided the architecture of our platform. So the online platform distinguishes three user domains which are specified in the following subsections.

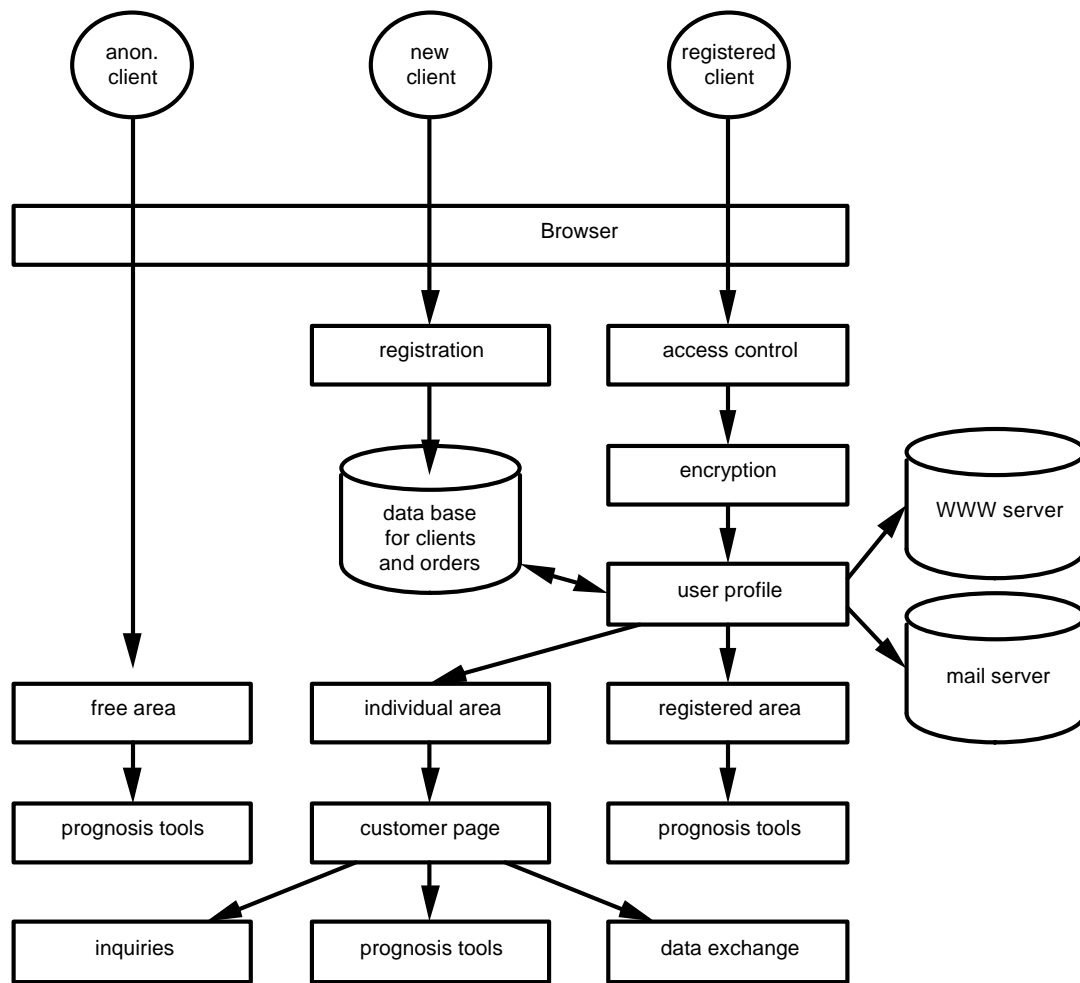


Fig.2: Architecture of the ePING server

4.1. Offerings for anonymous customers

This domain offers well-proven and established prognosis tools in ship design in order to demonstrate the functionality of ePING. Access is free of charge and meant as an initial offer for users who are from time to time faced with design tasks and search for quick and competent answers to their problems. We expect anonymous clients to keep in touch with the platform and change to registered users eventually becoming long-term customers. Data transfer between client and server is unencrypted and can be monitored by a third party.

4.2. Offerings for registered customers

For registered customers a password-protected working area with a limited amount of server storage is set up. Users may define in their personal profile in which language (English, German) they wish to communicate. Furthermore they can decide how results computed by available tools are handled: Shall data be deleted immediately at session end or be kept for future look-up or processing.

Registered customers receive additional benefits: encrypted data transfer between client and server, mail service, background information, access to extended features of tools (e.g. graphical editing of results), contact persons etc. Use of the registered domain is also free of charge but users have to leave some personal and corporate data (name, address, phone, email) which, however, are only used for server administration purposes.

4.3. Offerings for individual customers

Working in the individual domain is only possible after a written license agreement. Customers may use additional software applications or access SVA's data bases with report data, publications or hydrodynamic formularies and are charged a license fee.

The target group of this service are customers with a long-term business relation to SVA. For a limited time the user leases a software tool either in order to solve a particular problem, for instance, or to become acquainted with the product prior to buying or permanently leasing it from SVA.

5. Applications

5.1. Open water diagram for SVA high speed propeller series

In a recently finished research project systematic tests of high-speed propellers have been carried out in the towing tank and the cavitation tunnel at the SVA leading to polynomial coefficients in an analogous manner like Wageningen propeller series, *Heinke et al. (2003)*.

Open-water diagrams for high speed propellers without cavitation may be simply and quickly computed using the online platform. In the registered area of ePING a form has to be filled out with the required input data and sent to the SVA server. The server checks the data, computes the result and sends it along with the open water diagram back to the customer.

Open-water diagrams with further parameters may be computed in the individual domain. The method allows a calculation of high-speed propeller characteristics with variable propeller parameters as well as predetermined operation parameters like cavitation or oblique inflow. The server calculates the propeller characteristics for given propeller data and the optimum propeller for given working points.

5.2. Automatic tender preparation for standard model tests

If one carefully analyses the process of preparing a standard offer, one may identify activities which are time-consuming and error-prone, if done manually, but can be formalized and thus automated.

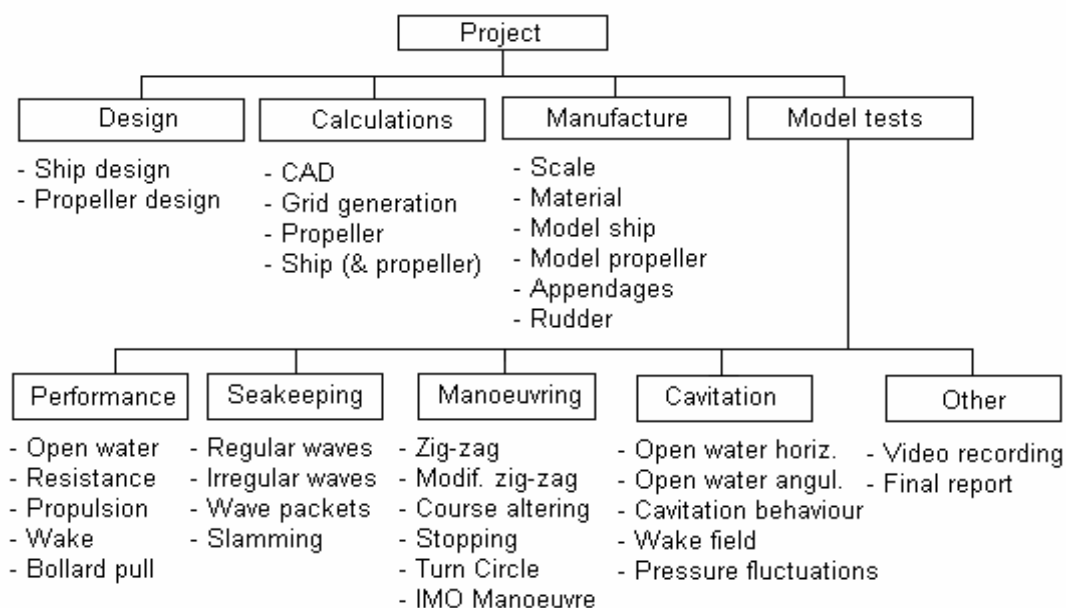


Fig.3: Backbone network for standard offers

Preparing an offer can be hierarchically subdivided into substeps depending on particular conditions and parameters. Possible substeps are defining the model size, searching for propellers in stock, number of model tests etc. Specification of a suitable model scale needs considering additional influencing factors: propeller diameter, milling cutter dimensions, towing tank geometry, experiment types etc. Different model scales imply variable production costs of the model. For an automatic price calculation it is essential to take these costs into account. The individual service items may be matched with the price data base and the customer data base and completed to a first standard-service offer. Of course, the price is preliminary and should be fixed in a counseling interview.

Currently, most of the procedures are used as often as possible by SVA staff members who with additional access rights to price information's can prepare formal offers.

5.3. Wave resistance prediction

While procedures with few input data are particularly suitable to be processed on the Internet platform, most ship design tools need detailed geometry descriptions, extensive parameter specifications and a considerable computation time. Moreover, for a fine ship grid a graphical interaction is required.

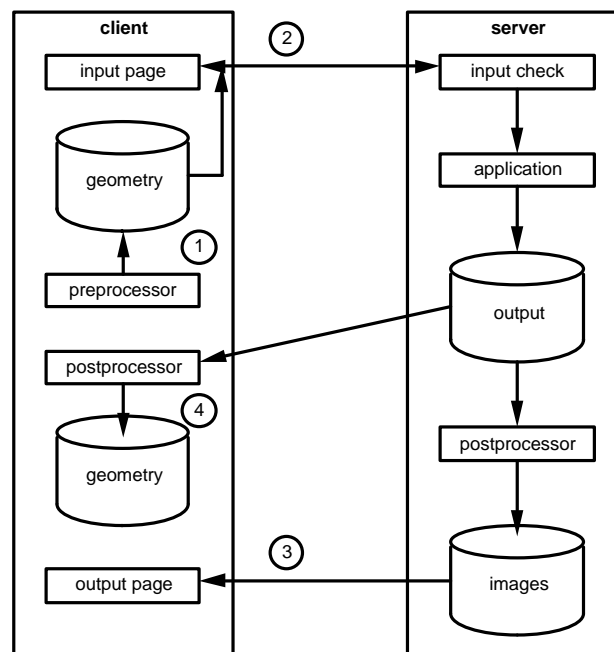


Fig.4: Simplified online calculation scheme

Fig.4 shows how to use the ePING platform in these cases. The customer applies a local preprocessor to convert his CAD data into a ship grid (1). He enters calculation parameters into a web form and sends grid and parameters over an encrypted connection to the SVA server (2). The server software checks the data and starts the computation. In his individual area the customer may look-up a progress indicator and, immediately after the calculation finishes, find a graphical report (3) or the full result in a file from which further images (4) can be generated using a postprocessor.

In general a customer would probably not appreciate such a procedure which seems complicated at first glance but rather he would order a full calculation. However, the process becomes useful if the customer has already computed results and wants to quickly repeat the calculation for only slightly modified operation conditions (e.g. different velocity) or for small geometry variations. In these cases the customer receives precise information faster and less expensive.

Basis of the wave resistance prediction method is the potential (Rankine source) code „Kelvin“, *Söding (1993, 2000)*. „Kelvin“ applies the „patch method“ to determine the wave resistance by satisfying the body and the nonlinear free surface conditions using sources distributed within the body and above the discretized part of the free surface. Fig.5 shows a typical wave pattern computed by „Kelvin“.

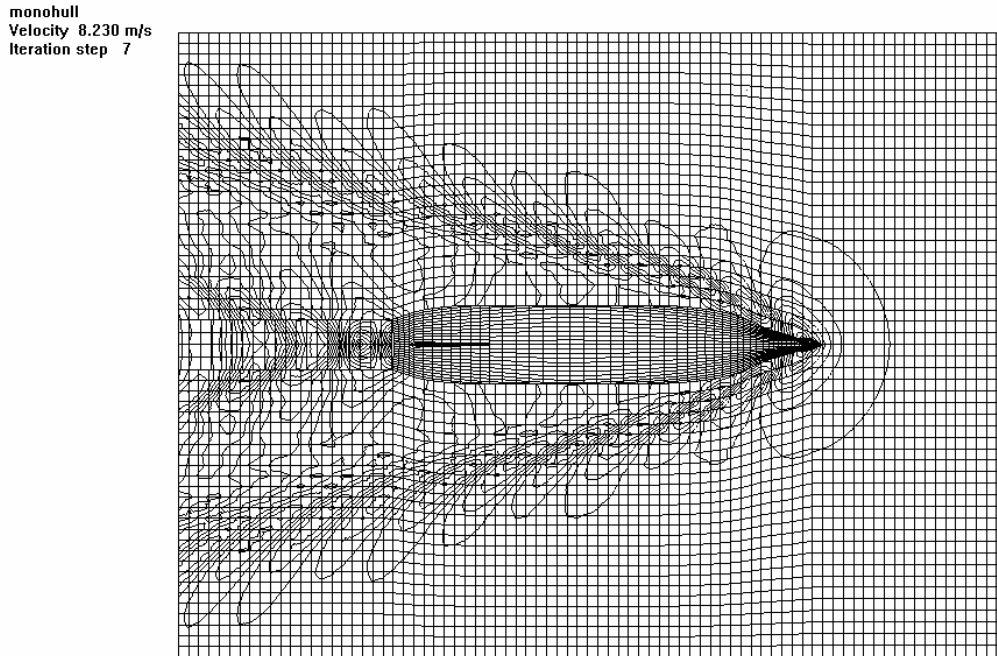


Fig.5: Wave pattern for a fast monohull

6. Data security

The online platform takes into account the current security standards. If necessary data will be encrypted in the registered and in the individual domain during the transport. All data are taken as confidential, are not passed to a third party and are only used for the intended purposes. The data are stored in a standard data base system and protected against incidental or deliberate manipulations, loss, destruction or access of unauthorized persons. Stored data can be retrieved or deleted according to the applicable law at any time. The security measures will be improved according to the technological development.

7. Technology and Tools

ePING is implemented on an Apache server with SSL, a firewall and a MySQL database. The web pages are written in HTML, Javascript and PHP. Incoming user data are redirected to the appropriate software tools and the results presented to the user. Cookies are used, if necessary, in order to store a session identification during the connection to the server. The online platform now runs for more than a year in a stable state and is permanently extended by new functions.

8. Concluding remarks

If members of the shipbuilding community succeed in using the Internet to offer prompt and customer-oriented engineering services on demand it will give them a considerable competitive advantage and guarantee their position in the market. By providing methods of calculation and hydrodynamic engineering knowledge on the Web SVA makes a first step in this direction.

Acknowledgment

This work was partially supported by the Ministry of Education and Research (BMBF) of the Federal Republic of Germany under grants no. 18S0216 and 18S0216A. The opinions expressed herein are those of the authors.

References

- KNICKEL, S. (1999), *Leitfaden E-Business*, PwC Deutsche Revision, Frankfurt am Main, p. 10
<http://www.pwcglobal.com/extweb/pwcpublications.nsf/docid/4A0F959287A087B285256BDC004393C8> (checked: 02-27-2004)
- DOMEYER, F. (2002), *Systematische Methoden und Verfahren zur automatischen Klassifikation von Dienstleistungen (Systematic methods and techniques for automatic classification of services)*, Department of Computer Science, University of Potsdam, Diploma Thesis (in German)
<http://www.informatikdidaktik.de/Examensarbeiten/Domeyer.pdf> (checked: 02-27-2004)
- FINKELSTEIN, C. (1999), *The Role of XML in Business Re-Engineering*, Information Engineering Services Pty Ltd., p. 2
<http://members.ozemail.com.au/~visible/papers/bri-xml.htm> (checked: 02-27-2004)
- HEINKE, H.J.; LAMPRECHT, M.; SCHULZE, R. (2003), *SVA High-Speed Propeller Serie*, Report No. 2943, Potsdam Ship Model Basin
- SÖDING, H. (1993), *A method for accurate force calculations in potential flow*, Ship Techn. Res. 40/4, p. 176-186
- SÖDING, H. (2000), *Program KELVIN – Methods and some results*, Report No. 2582, 8th SVA-Forum, Potsdam Ship Model Basin

Propulsion Plant Models for Nautical Manoeuvre Simulations

Tobias Haack, Stefan Krüger

TU Hamburg-Harburg AB 3-14, Hamburg/Germany, t.haack@tu-harburg.de

Abstract

A time domain simulation tool for manoeuvring purposes based on body forces has been developed, integrating the major features of the propulsion control system. Simulations for different propulsion concepts and ship types are compared with full-scale trial data. The characteristics of the nautical manoeuvres and the behaviour of the plant can be well reproduced. It is also possible to predict the system performance before going to the trial trip with sufficient accuracy even in off-design conditions.

1. Introduction

The simulation of nautical manoeuvres becomes most important for such ship designs where the manoeuvring requirements are much higher than the prescribed standards set by the IMO. This holds not only for those manoeuvres where interim guidelines exist, such as turning circles, zig-zag manoeuvres, or crash stops, but mainly for “Off-Design” situations such as crabbing. Especially for these types of nautical manoeuvres, but also for new propulsion systems such as podded drives, the behaviour of the propulsion plant as well as the details of the propulsion control system become a governing criterion for the behaviour of the whole system. Therefore, a time domain simulation tool for manoeuvring purposes based on body forces has been developed, integrating also the major features of the propulsion control system. Many simulations for different propulsion concepts and ship types have been performed and compared with full-scale trial data.

2. Model

For the numerical simulation of ship manoeuvres there are two types of basic models:

- Motion Models (or coefficient models) solve the equations of motion, which are formulated as coefficient terms. The relevant coefficients have to be determined in model tests for certain manoeuvres. The coefficients are then used to predict the motion of the ship in other types of manoeuvres. The disadvantage is manoeuvre trial or model data are required. Also the scaling of coefficients from model size to full-scale ship is problematic: (1) viscous effects especially at the appendages are overestimated and (2) the calculated coefficients are only valid for one draft (e.g. the design draft). This is why coefficient models are hardly suitable for practical ship design.
- Force Models sum up all forces acting on the ship at a certain time t and compute the resulting accelerations. Velocity and track are calculated by integrating the accelerations. The chosen method bases on a model developed by Söding (1984).

The following equations of motion have to be solved.

$$\begin{aligned} X_{hydro} + X_{Rudder} + X_{Prop} + X_{aero} &= m(\ddot{u} - v\dot{y} - x_G\dot{y}^2) \\ Y_{hydro} + Y_{Rudder} + Y_{Prop} + Y_{aero} &= m(\ddot{v} - u\dot{y} + x_G\dot{y}^2) \\ N_{hydro} + N_{Rudder} + N_{Prop} + N_{aero} &= I_Z\ddot{y} + mx_G(\ddot{v} + u\dot{y}) \end{aligned}$$

The implementation of these equations results in a tool that is modular in two ways: (1) It is possible to refine the different force components. For example, the data concerning the ship resistance can be put in concrete terms, starting from a first rough estimation and ending with the results of the resistance model test. (2) It is possible to activate or deactivate single force components (e.g. wind).

The focus within this paper is the integration of a new model for the simulation of the power plant and the propeller into a existing manoeuvre simulation software.

2. Propulsion Model

We discuss here only the new and relevant modifications of the propulsion model. Within the research project the propulsion variants shown in Table I were analysed.

Table I: Propulsion variants

2-stroke Diesel	4-stroke Diesel	Diesel-Electric
Fixed Pitch Propeller	FPP	FPP
Controllable Pitch Propeller	CPP	CPP
		POD with FPP

2.1. Propeller Model

The propeller forces are described by KT-KQ curves as far as these curves are known. These diagrams exist for many different propellers with different pitch, diameter, and disc area ratio. The coefficients for the wake and the thrust deduction are often known either from a vessel of comparison, direct calculations (CFD), or model tests. These diagrams are suitable for the simulation of ship manoeuvres such as turning circles or zig-zag manoeuvres to compute the propeller forces in the relevant velocity range. If no trial data of a comparable propeller is available, the Wageningen B-Series is used, *Van Lammeren et al. (1969)*. Problems arise if the propeller is not working in the Forward velocity / Forward thrust condition or if a CPP works in an off-design condition. These operating conditions occur during crash-stop manoeuvres and crabbing manoeuvres in the harbour. The implemented propeller model which overcomes this problem is explained below.

For the fixed-pitch propeller, the open-water curves of the Wageningen B-Series are used if no better data is available. In addition to the standard KT-KQ curves, open-water diagrams for the operating conditions Forward velocity/ backwards thrust, Backward velocity/ Forward thrust, and Backward velocity/Backward Thrust (Four-quadrant Model) exist, *Van Lammeren et al. (1969)*. The four-quadrant data of the Wageningen B-Series is also used for known propellers to describe their behaviour in the 3 non-tested quadrants. In this case, the Wageningen Curves are modified by a correction function to coincide the two curves in the first quadrant, Fig.1.

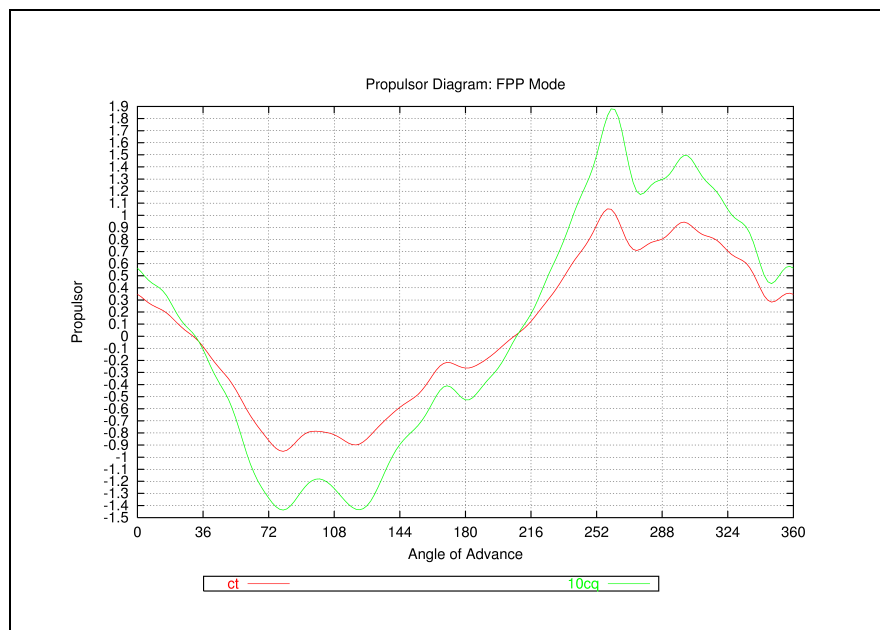


Fig.1: Open-water Diagram for a Fixed Pitch Propeller

Although controllable-pitch propellers (CPP) are often not operating with their design pitch, the open-water curves are normally only available for this condition (e.g. from a model test). This is insufficient for a manoeuvre simulation. Initially one may be tempted to use the Wageningen Series, because they include a wide range of pitches, but CPPs behave differently if operated with reduced pitch than an FPP of same average pitch. Therefore we used a propeller model which consists of numerically calculated open-water curves for different operating conditions (variable pitch and advance coefficients). If thrust and velocity have the same orientation, a vortex model is the method of choice. For the other conditions, a momentum theory (based on Glauert) is used, which includes the recirculation effects using empirical corrections. These computations are very time consuming. To minimize the computation time during the manoeuvre simulations, a couple of open-water diagrams are calculated in advance. Propeller thrust and torque are then quickly determined by interpolation between these curves during the manoeuvre simulation. In addition, correction functions are used additionally, Fig.2.

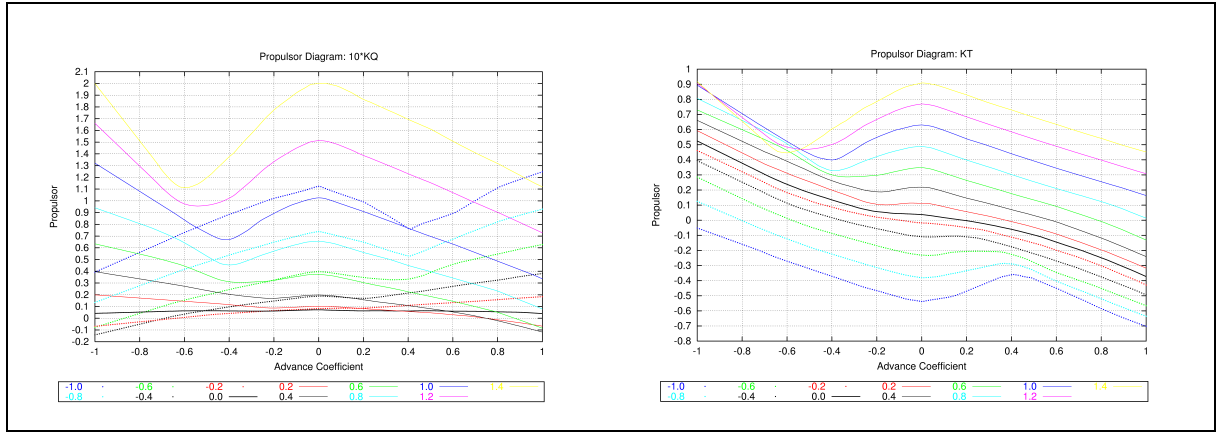


Fig.2: Open-water curves for a CPP

Pod devices are treated as fixed-pitch propellers in an angular flow, so that the angle of incident of the propeller corresponds to the “rudder part” of the pod, *Krüger and Urban (2000)*.

2.2. Propulsion Plant

Figs.3 and 4 show the model of the propulsion plant. Fig.3 shows the configuration of the implemented model of a 2-stroke diesel engine with a FPP. Fig.4 shows a 4-stroke diesel engine in combination with a CPP unit. The important parts of the power plant are described below.

2.3. Control System

The shaft governor in a FPP propulsion plant has to tune in the engine speed that corresponds to the lever position. In CPP propulsion plants, the governor has to hold the constant speed of the engine. In combinator mode, a combination of shaft speed and pitch corresponding to the combinator diagram has to be tuned in. The shaft governor is implemented as PI controller without reaction time. The following mathematical model is used:

$$\begin{aligned} torque &= p \cdot (n_{command} - n_{is}) + i \cdot n_{int} + torque_{start}, \quad \text{with} \\ n_{int, t_n} &= n_{int, t_{n-1}} + (n_{command} - n_{is}) \cdot (t_n - t_{n-1}). \end{aligned}$$

In addition, a hysteresis curve is implemented. This curve prevents the governor from reacting in the case of small deviations. Moreover, if a FPP is used, the shaft speed must not fall below the a certain shaft speed (idle speed) to prevent stall of engine.

The pitch controller tunes in the blade angle that corresponds to the lever. This is done by activating and deactivating the hydraulic pumps to control the blade turning rate. Furthermore, illegal operating

conditions have to be prevented. For example, it has to be ensured, that the engine is not wind milling during crash stop manoeuvres if the pitch is reduced too fast. An effective way to avoid this effect is to implement a crash-stop program which reduces the engine speed to idle speed. A combinator mode is also implemented in the simulation model. This means, that combinations of shaft speed and pitch value are adjusted according to the so called combinator curve as far as the combinator mode is integrated in the plant. The combinator curve includes the command values for shaft speed and pitch for the given lever command.

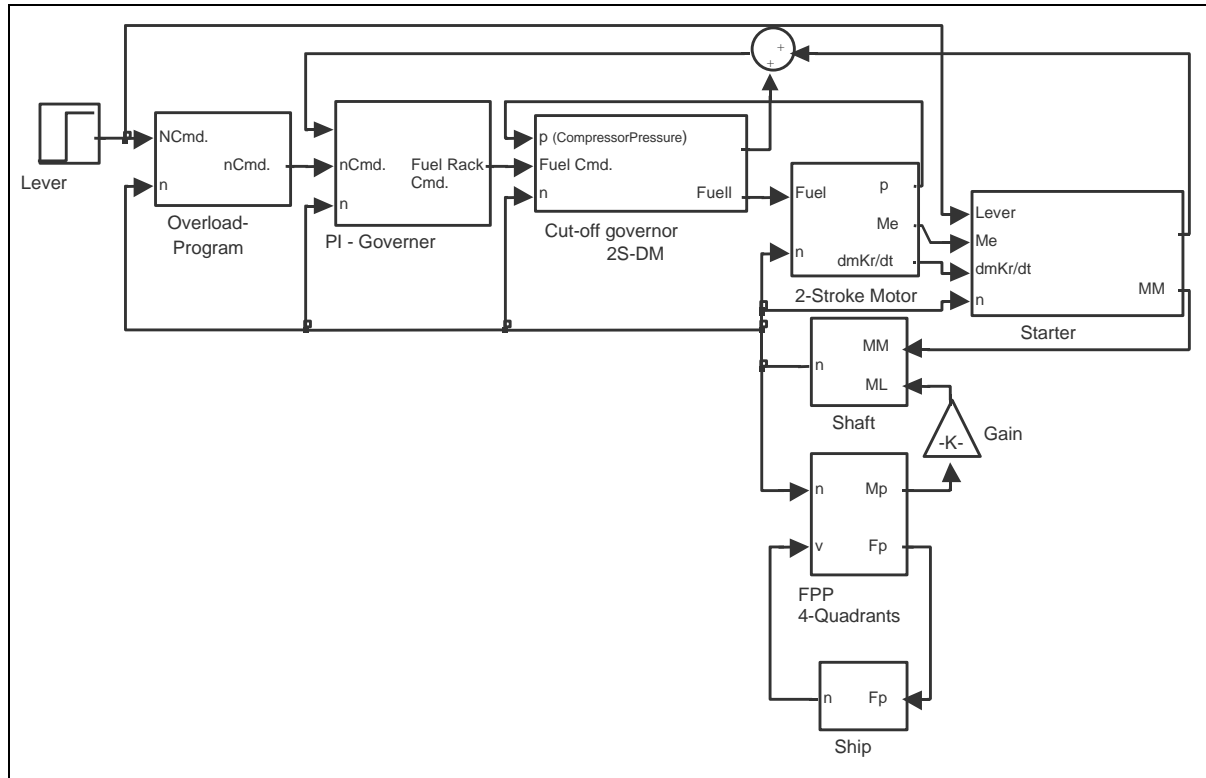


Fig.3: Block Diagram of a 2-stroke diesel engine with Fixed Pitch Propeller

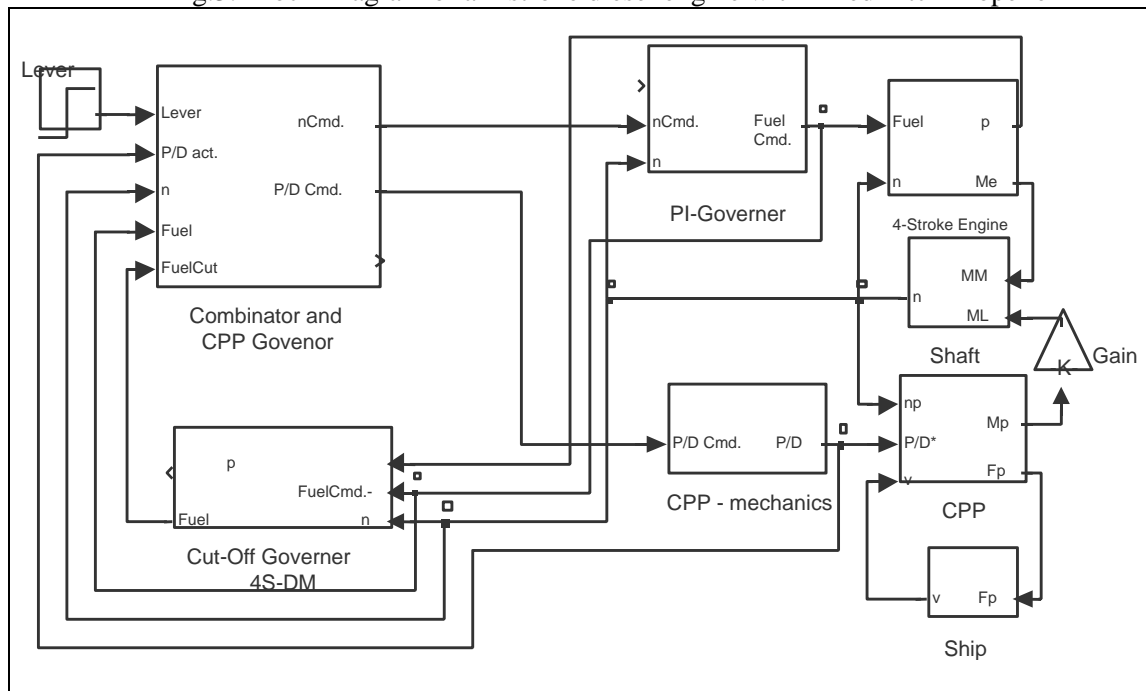


Fig.4: Block diagram of a 4-stroke diesel engine with Controllable Pitch Propeller

3. Examples

As test cases a 2 RORO-Ferries with different propulsion concepts were chosen. Ship 1 is a single-screw vessel with a 2-stroke diesel engine and a Controllable Pitch Propeller controlled by a combinator curve. Ship 2 is a twin screw vessel with 4-stroke diesel engines and Controllable Pitch Propellers in constant rev. mode. Different manoeuvres were simulated and the results are compared with the sea trial data below.

3.1. Turning Circle Simulation

Figs.5 and 6 show the good agreement between simulated and measured turning circle diameters. The difference (Ship 1: Tactical Diameter measured 541m; simulated 502m) are within the scope of what is reproducible during a sea trial. During the sea trial, the turning circle diameter is not only influenced by the environment (e.g. wind, sea state, shallow water) but by the actual state of the power plant. If for example the starboard turning circle is performed after the portside turning circle, it might be possible, that the engine has been overloaded and an overload protection program prevents the engine from providing the full available power. The pitch reduction is simulated precisely, Fig.6 (top right). The pitch of the outer propeller is reduced to 73%. The inner propeller has a pitch of 82 % of the design pitch, because the outer propeller is exposed to a greater cross flow than the inner propeller.

Ship 1

LPP	189.692 m
B	26.50 m
T _D	6.95 m
V	Approx. 20000 m ³
Service Speed	Approx. 23 Knots
Propulsion	2-stroke / CPP

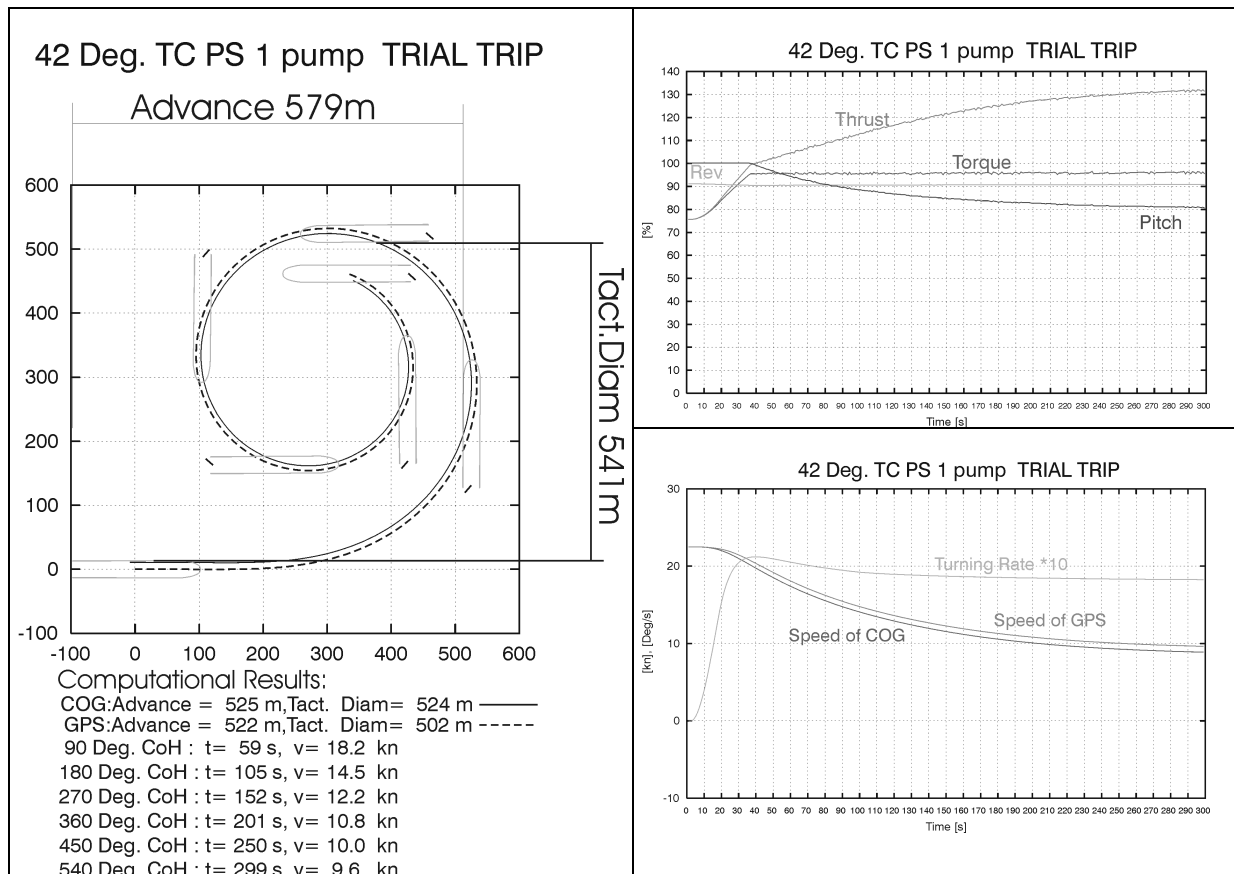


Fig.5: Turning Circle Ship 1: Track (left), simulated power plant (top) and speed (bottom)

Ship 2

LPP	182.390 m
B	26.00 m
T _D	5.7 m
V	Approx. 15,000 m ³
Service Speed	approx. 21.5 Knots
Propulsion	4-stroke / CPP

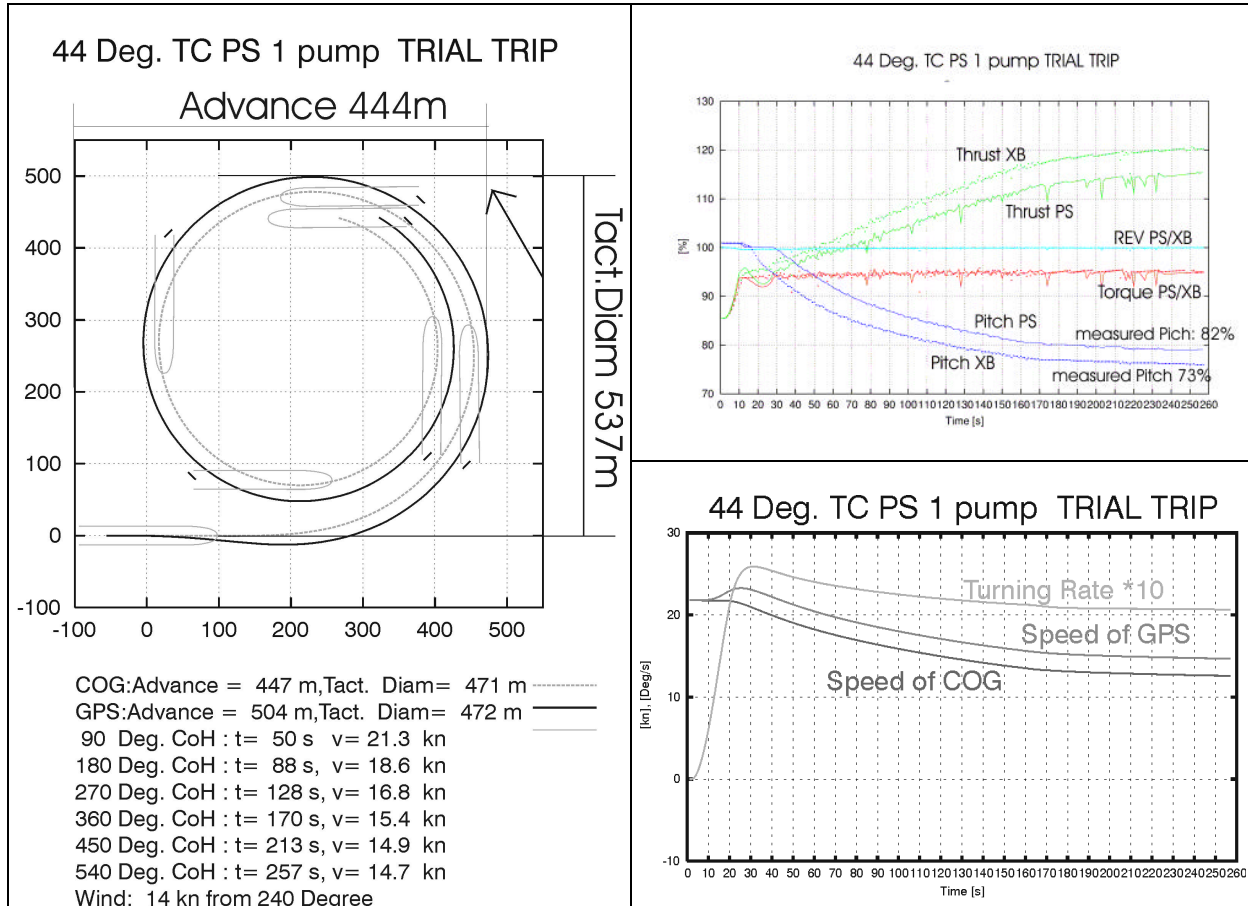


Fig.6: Turning Circle Ship 2: Track (left), simulated power plant (top) and speed (bottom)

3.2. Crash-Stop Manoeuvre Simulation

An other important manoeuvre is the crash-stop manoeuvre. The ship has to be stopped from maximum speed ahead. This is a crucial manoeuvre regarding the control units. Ships with power plants consisting of a 2-stroke Diesel Engine and a FPP are stopped by reversing the machine. So the stopping distance depends on the power of the starter (FPP) and the thermal protection unit. Ships with CPPs are stopped by reversing the blades. Figs.7 to 9 show simulated and measured data of crash-stop manoeuvres of Ship 1 (22.5 knots). Figs.7 and 9 show the result of the same manoeuvre with a different plant configuration. The plant diagram in Fig.7 shows the effect of the crash-stop program. The governor command is set to idle speed as the crash-stop command is given (lever command “full astern” from “full ahead”). This avoids the wind milling effect, and the maximum astern pitch is reached much faster. This control unit reduces the simulated stopping distance from 1620m to 1320m. Such a program is not implemented in the power plant of Ship 2. Between 50 and 70 second of the simulated trial (Figs.10 and 11) the wind milling effect is evident. The pitch can temporally not be reduced because the turbine condition of the engine has to be avoided. So the stopping distance is unnecessary increased.

Simulations and analysis of sea trial show that the blade turning rate of the CPP control unit does not influence the stopping distance significantly. Simulated and measured behaviour of the power plant agree qualitatively well. The calculated stopping distance unfortunately underestimates the stopping ability of the ship, possibly due to the approach for the values of the thrust deduction.

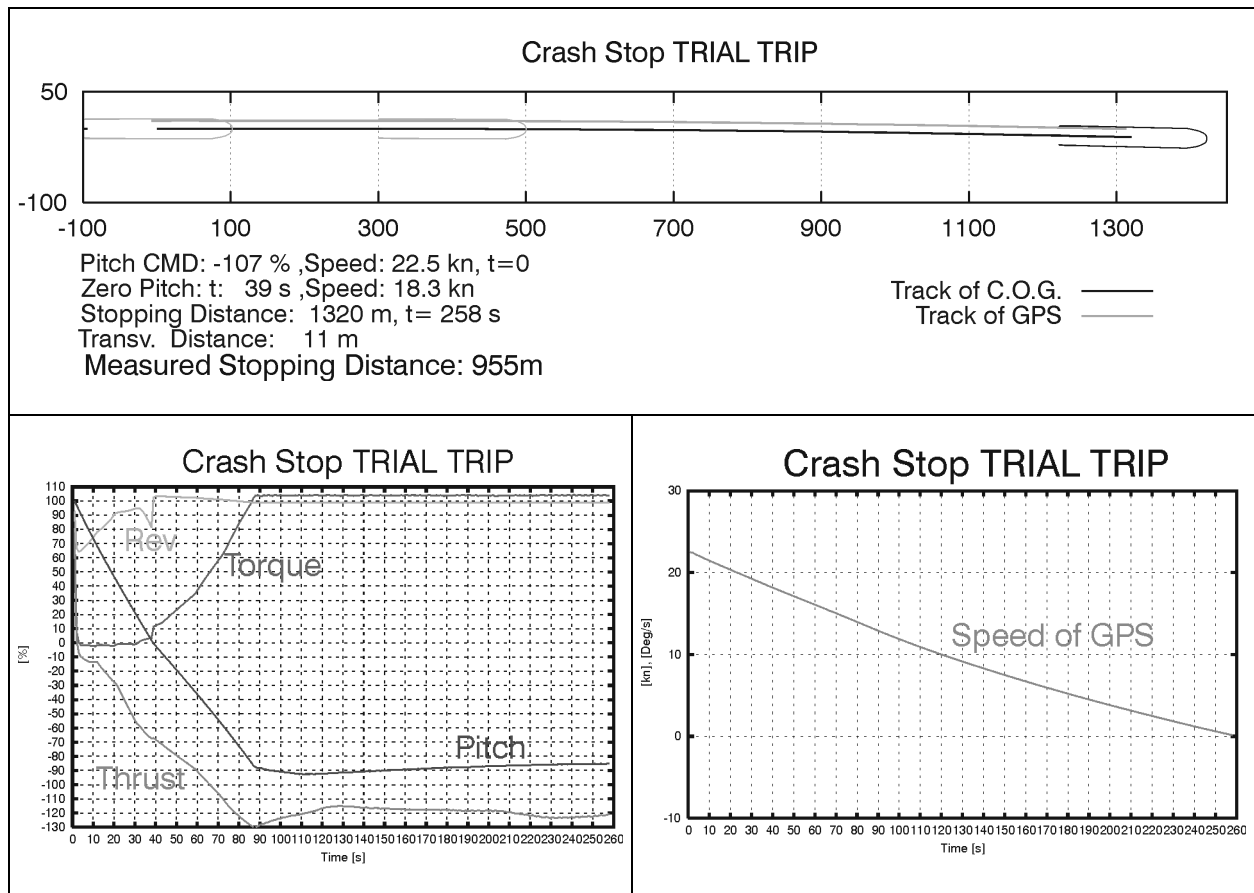


Fig.7: Ship 1; simulated crash-stop manoeuvre from full ahead (22.5 kn). Track (top), power plant (left) and speed (right); Crash Stop Governor Program is activated (corresponds to the sea trial)

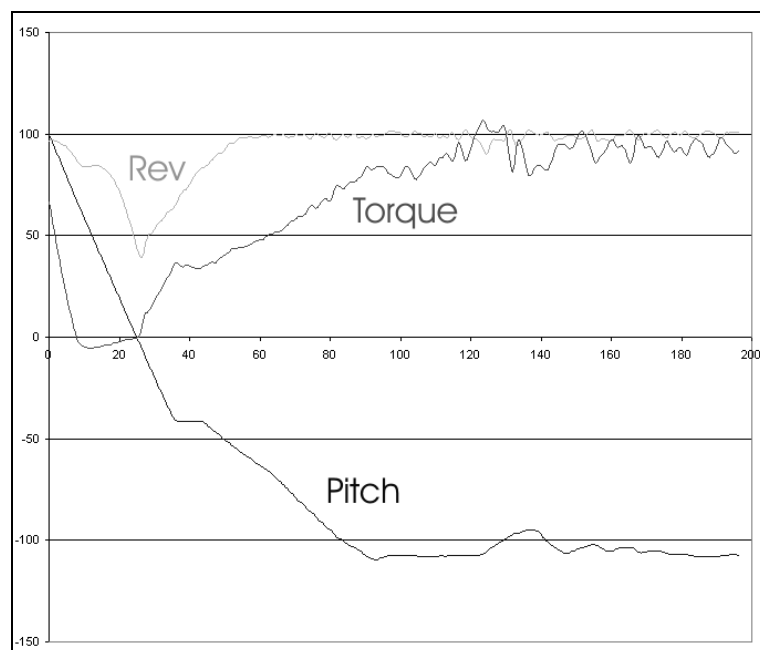


Fig.8: Ship 1; Measured power plant diagram in sea trial

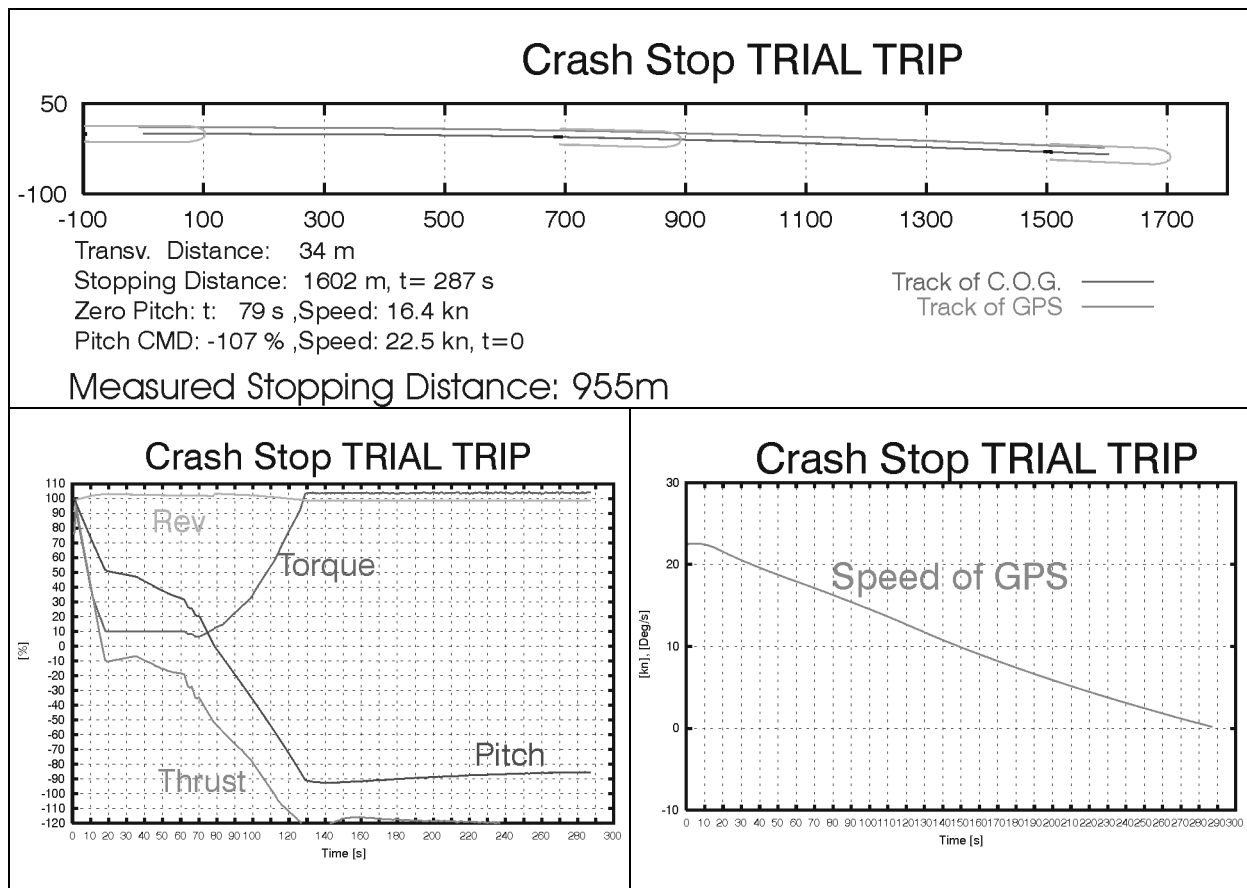


Fig.9: Ship 1; Simulated Crash stop manoeuvre from full ahead (22.5 kn). Track (top), power plant (left) and speed (right); Crash Stop Governor Program is not activated

4. Conclusions

The results of the manoeuvre simulations agree well with the measured values from the sea trial. The standard manoeuvres can be predicted sufficiently well. The simulation of turning circles, zig-zags and Williamson turns in the design process enables the engineer to evaluate whether his ship fulfils the requirements of the IMO and the customer without making expansive model tests. He can also estimate how the controllers have to be adjusted to guarantee the best possible manoeuvring performance. The results from the crash-stop simulation differ from measured data for several reasons. There is no knowledge about the thrust deduction during the crash stop manoeuvre. An extension of the current work shall implement a direct computation method to overcome this problem. This will allow then to compute thrust deduction coefficients in all four quadrants.

Furthermore the simulation model will be integrated into different systems for nautical training in the near future. These systems with their visualisation tools will allow to use the manoeuvring model interactively. Therefore nautical instruments of different manufacturers have to be implemented.

References

- KRÜGER, S.; URBAN, A. (2000), *Abschätzung der Manöviereigenschaften von Azimuth Antrieben in der Projektphase*, Handbuch der Werften XXV, Hansa-Verlag
- SÖDING, H. (1984), *Bewertung der Manöviereigenschaften im Entwurfsstadium*, Jahrbuch der Schiffbautechnischengesellschaft, Springer
- VAN LAMMEREN, W.; VAN MANEN, J.; OOSTERVELD, M. (1969), *The Wageningen B-Screw Series*, Trans. SNAME 77, pp.269-317

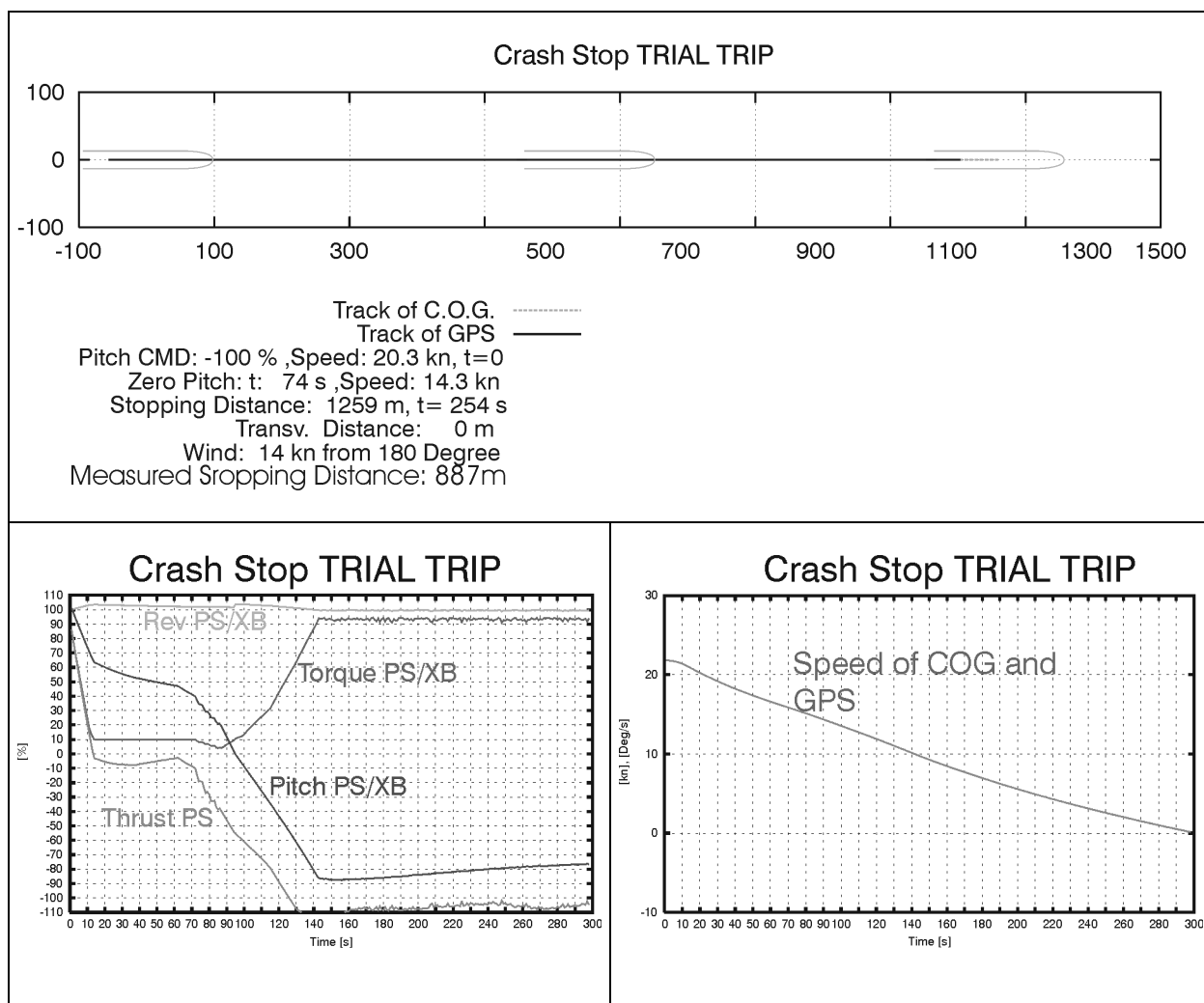


Fig.10: Ship 2; Simulated crash-stop manoeuvre from full ahead (20.3 kn). Track (top), power plant (left) and speed (right); Crash Stop Governor Program not available

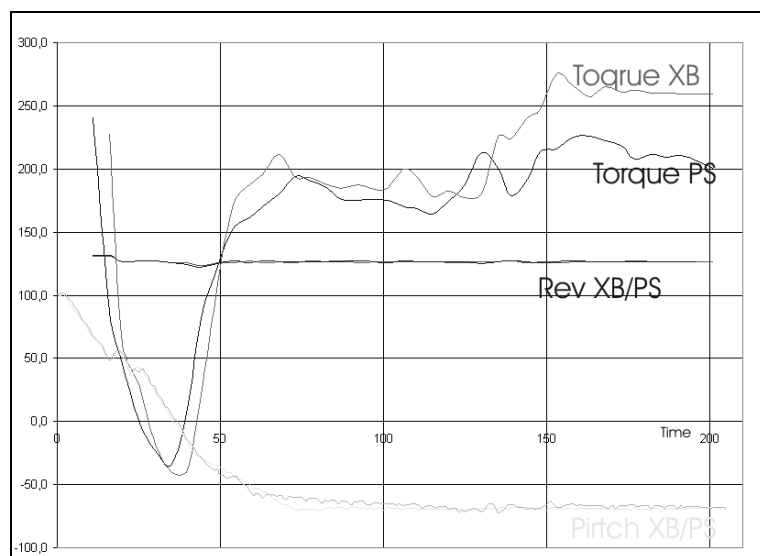


Fig.11: Ship 2; Measured power plant diagram

Using Simulation in Evaluating Berth Allocation at a Container Terminal

Lawrence Henesey, Paul Davidsson, Jan A. Persson Blekinge Institute of Technology,
Karlshamn and Ronneby/ Sweden, lhe@bth.se, pdv@bth.se, jps@bth.se

Abstract

The operations and decision making at a container terminal have been simulated. A Berth Allocation Management System – (BAMS) has been built which consists of two parts: a container terminal simulator modelling the operations and a management simulator modelling the various actors involved in the allocation of container ships to berths. Together these two parts generate berth schedules for arriving container ships. Two berth assignment policies are evaluated in different scenarios, with various quay lengths, berth spacing lengths, and ship arrival sequences. The decisions in assigning ships with different loading and discharging demands to a limited amount of resources, such as berth space and cranes are analysed with the BAMS. The berths at the container terminal are modelled by the BAMS to be dynamic in the sense that berth segmentation is based on the current situation rather than being static. The policies are evaluated in terms of turn-around time and distance travelled by the straddle carriers. The simulation results indicate that an informed choice of berth assignment policy can provide better use of the available resources, e.g., by reducing turn-around time and/or distance travelled by the straddle carriers.

1. Introduction

This paper investigates the use of simulation as the basis for a decision support system in analysing the assignment of berths to arriving container ships at a CT (Container Terminal) under various constraints and policies. The underlying aim of the research has been to increase CT performance without physical expansion by utilizing the available resources more efficiently. Through CT visits and video taping the operations and the managers, policies that were used in the daily operations have been codified, simulated, and analysed. The CT simulator and the management system are unique in that the decision processes used in the planning of the operations are recognized, modelling the CT as a system of decision makers. The proposal is to assist CT managers in the assignment of container ships to be berthed by using BAMS (Berth Allocation Management System) as a part of a Decision Support System (DSS). The BAMS consists of a simulated representation of a CT (cranes, berths, quays, transport equipment, containers, and ships). In addition, the BAMS includes a management system that represents the managers by issuing and sending documents (i.e. ship schedule, resource schedule, waiting time, crane schedule, etc.) Together, the two parts of BAMS assist in creating berth schedules for arriving ships under various conditions.

The aim of the BAMS is to efficiently use the resources available during the operating time that the container ship is occupying the berth. According to *Nishimura et al. (2001)*, the berth allocation plays a primary role in minimizing the *turn-around time*, the time that a container ship is worked, because the container ships being berthed do not have the same handling times. Turn-around time is one of the main performance measure used in port operations. Different berthing points may influence the handling time and distances being travelled by the transporters in order to “work” a container ship. According to Eric D’Hondt at HesseNordnatie in Antwerp, Belgium; “...as container ship size increases, the berth productivity becomes ever more important to ensure that the container ships can adhere to their sailing schedules.” The CT managers interviewed all mentioned that berth planning is a complex task and is critical before container ships can start the loading and discharging operations. Majority of the managers interviewed, interestingly mentioned that they do not use any tools in the assigning of container ships to berths, but have developed routines, contracts and “rules of thumb”. All of the managers agreed that they would benefit from having a decision support tool like a BAMS to help determine berth allocation and analyse their decisions or *policies*.

The application of discrete event-driven simulation where there exists queuing and scarcity of the number or availability of resources is viewed as a valid approach in simulating a CT *Bruzzone (1999)*. The CT is seen as a system and its 'state' changes according to discrete points of time, which can later be simulated for evaluation. We believe that the use of simulation by CT managers in solving the berth assignment problem, such as considering the impact of yard configurations may assist CT managers in developing interesting solutions, i.e. assigning incoming containers to stacks locations in the CT according to type, ship, or destination. As mentioned by *Ojala (1992)*, simulation used in connection with Decision Support Systems can be beneficial to port organizations. According to *Leathrum and Karlberg (2000)*, the primary purpose for the use of simulation is to determine the necessary resources in order to complete the processes (e.g., loading and unloading a container ship) within certain constraints. Though much research has concentrated on optimising the resources at the operational level, this paper seeks to examine strategic planning and tactical decisions that are made by the CT managers. Many of the decisions that are made in the execution of the CT operations are generated from set plans, established layouts or procedures and policies. The simulation of assigning berth with the BAMS is not expected to provide an optimal berth solution; it does provide an interesting means of testing and evaluating berthing policies.

In the next section the background and problem is described. This is followed by the presentation of the model and experiments performed with BAMS. Finally, conclusions and pointers to future work are provided.

2. Background

The base model for the CT is the Skandia Harbour (SH) located at the Port of Gothenburg, Sweden, which is approaching its current per annum capacity, 750 000 TEU (twenty-foot equivalent unit steel container). There has been plans to increase the capacity through physical expansion and has recently been awarded 450 million SEK by the Swedish government in 2003. The methods to increase CT capacity of containers when limited to the present physical size are changes in types of equipment, yard and terminal layout, and policies. Currently, the capacity level target has not yet been decided

A preliminary data collection analysis of thirty international sea ports and CTs invoked several questions, such as whether quay length and number or size of berths have an influence to ship turn-around time and/or throughput of containers, (analysis showed that there is no direct correlation solely between quay length and throughput, please refer to Appendix 1.) Much data and information is difficult to obtain, such as ship turn-around times at the various ports for a given year. Total quay length and number of berths was collected and derived from Containerisation International and from the current research work in order to evaluate the number of TEU per m of quay handled, a widely used industry standard for benchmarking CTs. There appears to be great differences with respect to throughput in relation to size of the quay and the number of berths, which may indicate the potential of increased utilization by for example improved policies. The number of TEU per m handled specifically motivated the research on berths, e.g. Hong Kong and Singapore have similar quay lengths but different number of berths (refer to Appendix 1). The reasons for these differences are not covered in this paper but do provide background to the problem, e.g. can alternative methods to satisfying customers be realized without having to increasing berth length and add resources. Most of the CT managers interviewed stated that fast CT operations at a low cost were demanded by the customers and in turn effected the management of the berths and the rest of the CT.

3. Problem Description

For the customers of a CT (owners of the ships and the shippers), it is paramount to minimize *turn-around time*, i.e. the waiting, loading and discharging of containers should be done as quickly as possible, in order to save on terminal costs. According to *Kia et al. (2000)*, an average container ship spends nearly 60% of its time berthed in a port at a current daily cost of \$65000 or more for a larger

ship. To shorten the time spent by container ships, terminal operators spend special emphasis in resource allocation and berth assignments to ships. The objective of the solutions employed by the observed terminals is to increase capacity at the CT by reducing the turn-around time. Some solutions used are:

1. Increase the length or number of berths
2. Increase the productivity at the berth by acquiring new technologies or machinery
3. Increase the time that berths can be operated (i.e. 24 hours, 7-days a week)
4. Improve the efficiency in allocation of resources
5. Improve policies and management decisions, which often have non-optimal objectives

In this paper the potential of the last solution will be investigated.

The major factors influencing both berth occupancy rates and turn-around time are: the number and size of arriving container ships; configurations of containers in the ship's bays; number of cranes; length of the berth and navigation constraints. Usually berth occupancy is based on the length of a container ship and the time it spends at the berth. However, high berth occupancy may result in congestion where container ships are queuing to be served, which would lead to high turn-around times equating to bad service for the container ships.

In the berth allocation task, each container ship that arrives at a terminal is assigned a berth and a location where it can dock in the terminal usually by a 'port captain'. The port captain has to balance two major objectives, ensure that a high berth occupancy rate is achieved and that the arriving ships are serviced according to the ship's demands. The port captain must also take into consideration that the resources assigned to the berth are allocated efficiently. The CT managers indicated during the interviews that in berth planning, placing the ship closest to the target stack on a first-come-first-served-basis was the method most commonly employed for berth allocation. From a strategic view, *Frankel (1987)* states that "The objective of berth planning by evaluation of congestion and cost as suggested by Nicolaou is to arrive at an optimum port capacity while incurring minimum capital cost". Thus, many CT managers are keen to minimize turn-around time with the resources that are available. CT managers are aware that in the "foot-loose" industry of shipping, that if customers are not satisfied they will seek other ports. There is a trade-off between the positive effect of serving ships faster and the substantial costs associated with investments in resources. Investments may for instance concern the berths, straddle carriers (SCs), and gantry cranes. Usually, every gantry crane will be served with a fixed number of transport machinery, i.e. SCs, which can transfer the containers in the terminal and can stack them to a certain height depending on the type of transport machinery employed and/or policies dictated by a yard manager.

4. The BAMS model and Policies

The decision to build a CT simulator stems from the limited amount of available working CT simulators to conduct research experiments. The model of the CT system built is an abstraction of a real world CT handling over 50,000 TEU per annum. According to *Jeffery (1999)*, CTs with more than 50,000 TEU per annum require an information system to help manage the processes and operations. We believe that BAMS would be a useful tool for such CTs where large numbers of containers are handled and CT managers need assistance in developing berthing decisions.

In building the CT model, input to the simulator was collected and data synthesized from Skandia Harbour with additional data and information from Norfolk International Terminals (NIT) in Norfolk, US and Seagirt Terminal (SGT) in Baltimore, US. These ports provided assistance and data in developing the configuration files of actual container ships to be simulated in the model.

As illustrated in Figure 1, BAMS consists of two components: the CT simulator modelling the physical entities and the management simulator modelling the managers. The decision processes of

the management simulator are sent as requests to the CT. The CT sends its state to the management simulator so that the simulated managers can develop a berth assignment schedule.

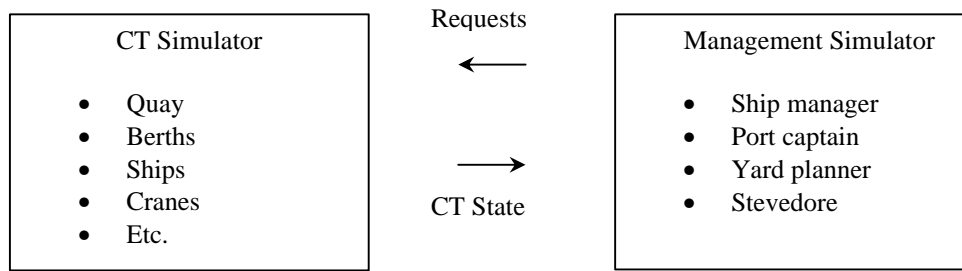


Fig 1: Simplified view of the BAMS model

In the management system simulator, a *port captain* is responsible for the allocation of resources at a dynamically changing part of the berth. Upon the request by a *ship manager*, the *port captain* assigns the berthing location of a container ship to a series of berth points. The *ship manager* sends the arrival time of a ship (ship schedule) for a container ship, a loading list and manifest (list of containers to be discharged) to the *port captain*. The *stevedore* provides a “snap shot” of the resources available to the *port captain*, additional information such as: capacity, cost, and type are included. In addition, the *yard planner* sends the properties of the yard stacks (height, position, type) to the *port captain*.

The number of cranes assigned to work the container ship is fixed and determined by input data. In the model, two cranes are allocated to each ship. Also, the number of straddle carriers are fixed and connected to a particular crane to reflect the real-world practices set by various dockworker unions (the number of straddle carriers assigned to a crane varies from three, e.g. SH to six, e.g. NIT and SGT). There are a number of additional aspects that effect the berthing assignment, such as:

- The configuration of the stacks in the yard.
- The number of containers and their configurations in the bays (stacks) of a ship.
- The number of available berthing points are varied, e.g. 1000 for a 1000 m quay.

The time that a ship waits to be served is called *Waiting Time* and the time to be served is called *Service Time*. In order to have a low turn-around time, both *Waiting Time* and *Service Time* must be kept minimal. Both the *Time* and *Service Time* are effected by the berth assignments.

The *Service Time* is generally known through the following computed information, which are classes (documents):

- *Ship Schedule*: The estimated time of arrival and departure for a ship as sent by the *Ship Manager*.
- *Berth Schedule*: The time needed to work ships are calculated and the distance to the stacks for each container.
- *Gantry Crane Schedule*: Set the allocations for the available cranes by calculating last position, time available and waiting time.
- *Resource Schedule*: Reports which machines are available and times that they can work.

The *Waiting Time* is calculated during the simulation, from a potential set of berth points that a ship in a system of ships would occupy. The number of possible of berth points depends on the berth spacing as well as a ship’s length plus a buffer distance. The ship *Waiting Time* may include time left in serving another ship that is occupying a part of the quay.

The computation of the *Service Time* is based on the number of SCs employed, the routes covered by SCs and their average speed. A route is calculated from the Berth Point along the quay, x to the far left position of a stack, z (see Fig. 2.). The routes are measured in meters. The sums of all the routes

travelled by each SC are totalled to provide the distance being covered by the SCs for each ship.

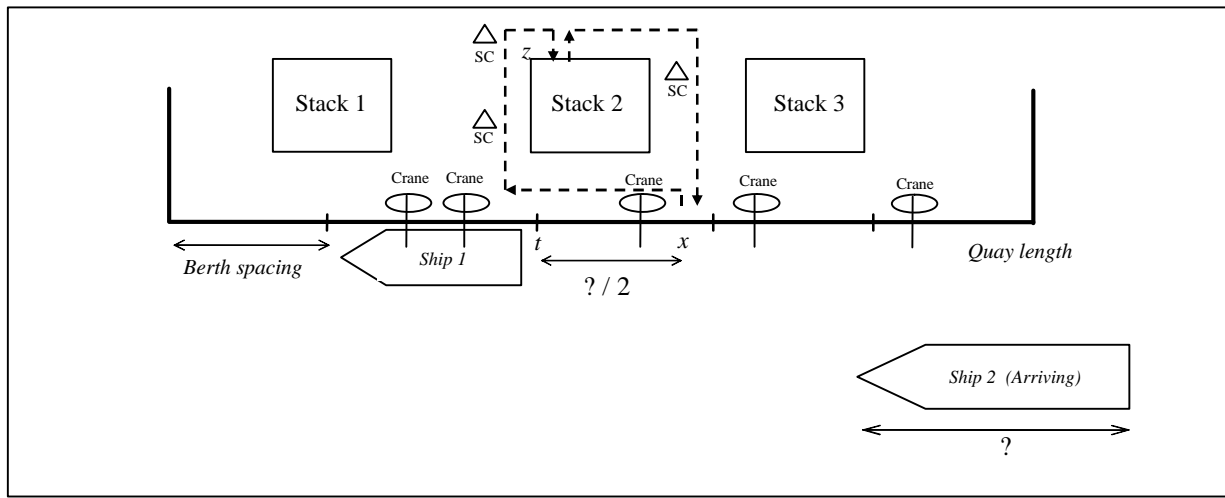


Fig. 2: The entities and distances involved in the assignment of a Berth Point.

Two berth allocation policies are evaluated: Berth Closest to the Stack policy (BCSP) and the Shortest Turn-around Time policy (STTP). The policies are used in the simulation experiments to determine the following:

- Where a container ship should be berthed?
- When should a container ship be serviced?

The BCSP places a ship closest to a ‘target’ stack. The target stack is the stack that will be most visited by the SCs during the operations. The BCSP will wait until a berth that is closest to the stack is available. The STTP objective is to place ships to berth positions in order to minimize the total turn-around-time for all arriving ships. In addition to *Service Time*, the STTP is considering the *Waiting Time* when determining the berth point for a ship. From the sum of the *Service Time* and *Waiting Time*, the STTP will place a ship wherever the shortest turn-around-time is achieved. The STTP can be characterised as considering the time dimension, where as BCSP is considering the space dimension when determining to place the ship.

The CT simulator was developed in the Java programming language and will be used for additional research, e.g. the implementation of agents in the BAMS and the market-based approach. The number of variables used for the configuration of the CT is over 50. The most important variables include: quay length; berth spacing; CT capacity; utilisation factor of the gantry cranes and straddle carriers; and yard stack positions and types. The communication between the CT simulator and with the management system is facilitated by the RMI facility of Java. The managers that are seen as objects in the software program are the following: *ship planner*; *stevedore*; *yard planner*; *ship manager* and *port captain*.

5. Description of the Experiments

The purpose of the experiments is to evaluate policies under various conditions. The manager system produced berth assignment schedules using the two policies described above for each of the case scenarios (described below), 5 quay lengths and 4 berth spacing allotments. Thus, the total number of berth assignment plans generated and examined was 120. The number of ships is 6 and their physical characteristics (lengths, number of bays, and container characteristics) were configured into each case. The ships are being worked on first-come-first-served basis

The input to a simulation experiment includes the following:

- *Policy*: Shortest Turn-around Time Policy (STTP) or Berth Closest to Stack Policy (BCSP).
- *Sequence of arriving ships*: The differences between the configuration files were in the arrival time intervals of the 6 container ships (4 x 260 m and 2 x 105 m) and the number of containers of the three different types, e.g. reefer, hazard, and standard, to load and discharge from each bay. The three cases were based on data provided by NIT and SGT. A general description is given below and the full details can be found at <http://www.ipd.bth.se/lhe/Simport>:
 - *Case 1*: The case is considered a best case because there is no ship congestion at the berth, thus, no need to decide in placing a ship into a waiting queue or locating an alternative berth. No two ships loading/unloading at the same time will “demand” the same berth location. The intervals between arrivals are set for the first ship to be served before the second one arrives.
 - *Case 2*: This case is characterized by CT managers as an “average case” in that some conflicts arise (minor ship congestion). Only one ship of the six has to be given an alternative berth.
 - *Case 3*: This case is characterized by short time intervals between the arriving container ships resulting in a queue of ships. A ship will typically have to wait in order to be berthed closest to the target stacks immediately, because the berth may be occupied.
- *Quay length*: The length in meters along the CT that is able to serve docked container ships tested at 400 m, 600 m, 800 m, 1000 m and 1200 m.
- *Berth spacing*: The spacing in meters between segments or berth spots along the berth from increments of 1 m, 100 m, 200 m and 300 m.

The output from the simulation experiment includes the following:

- *Berth assignment plan*: Schedule for assigning ships to berth points along the quay.
- *Crane Assignment*: Which cranes are assigned, their starting and final positions.
- *Service time*: The total time turn-around time, measured in hours, for all ships.
- *Distance*: Measured in meters the distance travelled by straddle carriers in order to move the containers (the ones to be discharged) from the gantry cranes to the yard stacks and to move the containers to be loaded from the yards stacks to the gantry cranes

Thus, the simulation experiment compares each policy (STTP and BCSP) for the three cases, where various quay lengths are tested with different berth spacing lengths to determine the berth position for arriving ships. The user of the simulation is able to produce a berth assignment plan for all arriving container ships based on either policy. The performance is measured in turn-around time and SC distance travelled.

6. Results

The main objective is to evaluate which policy is best under various conditions for each case. In Figure 3, the simulation experiment results compare the two policies under 5 quay lengths with 1 m berth spacing. The results are summarized from Appendix 2. When Case 1 had the quay set at 400 m the results of the simulation showed that distances travelled by the SCs are the longest and the turn-

around time is the highest against the other 4 quay lengths. As the quay is extended, the distance and time are decreasing for both policies until at 800 m. When the quay length is set at 800 m or longer for Case 1 the results are similar. In Case 1, where the traffic is low, the results indicate that there is no significant difference between choices of policies.

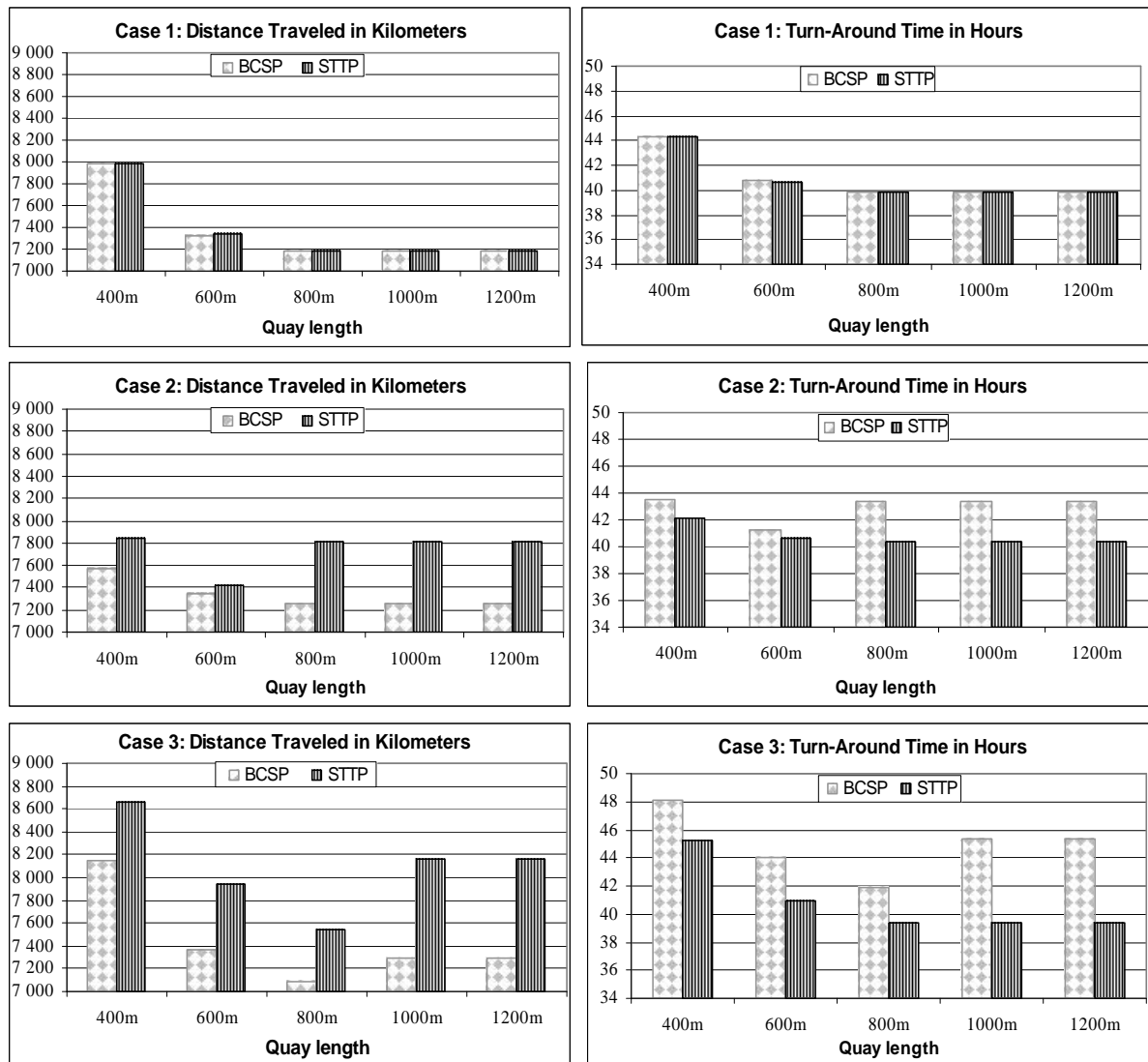


Fig. 3: Simulation Results in Experimenting Policies with Different Quay Lengths

In Case 2 and Case 3 the difference in the results between choices of policies is more apparent. In Case 2, the BCSP is resulting in shorter distances travelled then the STTP. In turn-around time, the STTP is better then the BCSP. The BCSP's best results for distances travelled are from 800 m to 1200 m. As the quay is lengthened the BCSP has ample capacity after 800 m to place ships closest to the target stacks. Thus, after 800 m the simulation will have similar results at 1200 pointing to overcapacity at the CT. The STTP's shortest distances travelled are when the quay is set at 600 m. The STTP is placing the ships to the berth based on turn-around time, when the quay is set at 600 m the number of ships placed at the quay is a maximum of 2. As the quay is lengthened, the number of ships to be given berths increases, however there are moved further from the stacks and are thus, increasing the distance travelled. In turn-around time the BCSP had its lowest turn-around time at 600m and STTP had its lowest turn-around time from 800 m to 1200 m. The BCSP is utilizing the resources much more than the STTP in order to have a fast turn-around time.

The most acute differences between BCSP and STTP are found in Case 3. As one would expect, BCSP is better with respect to distance travelled by SC, where as STTP is better with respect to time. For the STTP in Case 3, the best results for distance travelled were at 800m and for BCSP, it was from 800 m to 1200 m. In the best turn-around results for Case 3, BCSP was fastest from 800 m to 1200 m and for STTP, best time was at 800 m. The distance travelled when the quay is 800 m are much less for the BCSP than the STTP for the turn-around time the BCSP is achieving a faster turn-around time at 800 m quay length. The difference is attributed to the characteristics of the ships, e.g. ships having different lengths, number of bays and all are loading or unloading the majority of the containers to one stack.

In addition to testing the two policies under various quay lengths, the experiments with varying berth spacing were also conducted and analysed. The results are summarized in Figure 4. The choice of policies at 400 m has no influence on the output; basically 400 m is too small a quay to serve 6 ships in one day. Of the container ships arriving, four ship are set at 260 m in length, thus at a small berth there will be congestion for the CT under both policies.

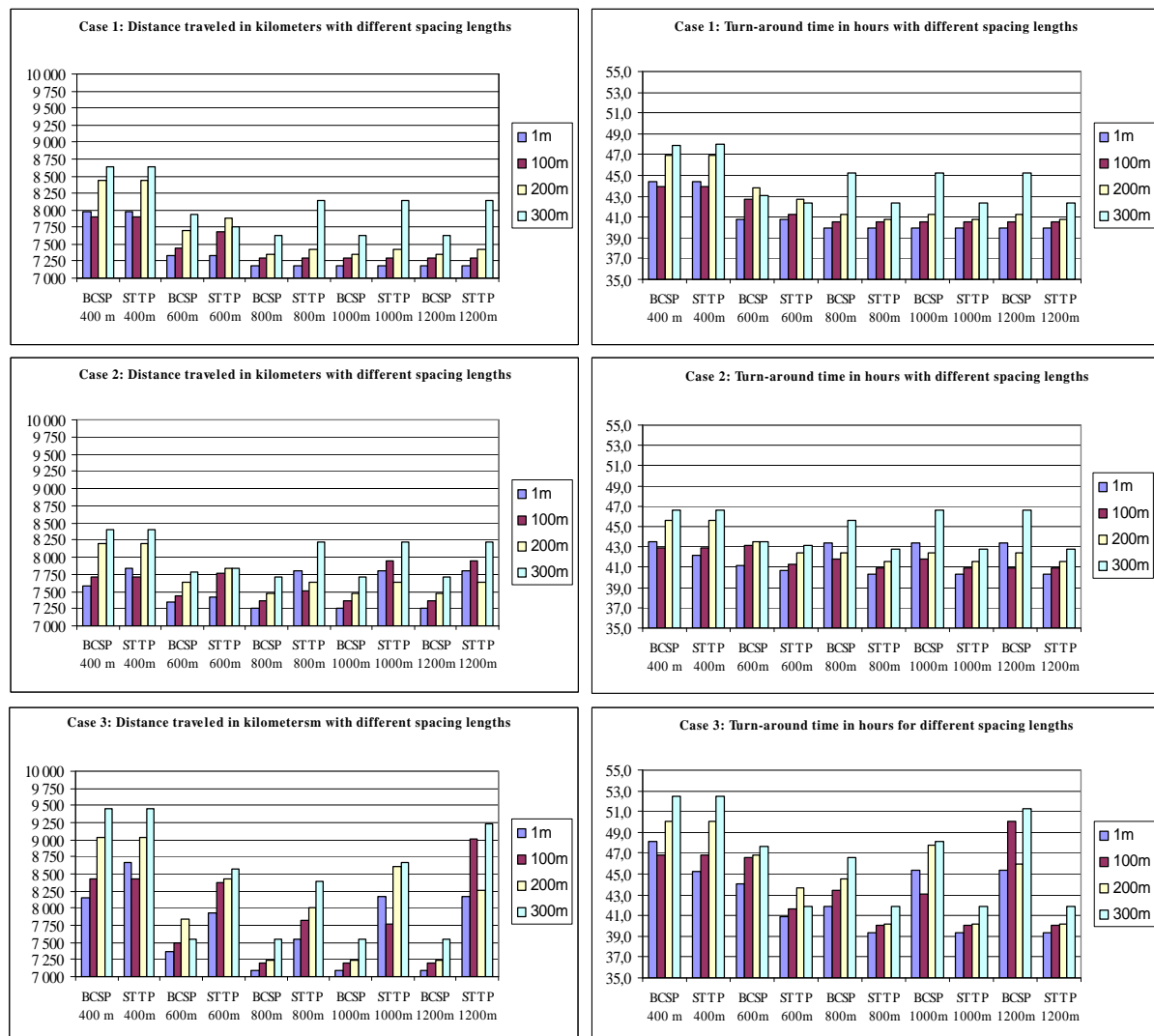


Fig. 4: Simulation Results in Experimenting Policies with Different Quay & Berth Spacing Lengths

In experimenting the two policies with different berth lengths and berth spacing (100 m, 200 m, and 300 m), the simulation results in Case 2 and Case 3 differ widely. The CT managers may want to change policies under those two Cases in order to realize their objectives, i.e. BCSP would result in less distance but higher Ship Service time. With respect to Case 2 and Case 3, a port captain seeking to reduce turn-around time may view the STTP as the best choice, so that the ships will be positioned

to alternative berths instead of placing them in a queue. The draw back to the STTP policy is that significant distance will be travelled by the SCs.

In Case 2 and Case 3, the BCSP indicated that the distances travelled by SCs are lower compared with STTP. However, ship service time is increasing, which may lead to further waiting time or congestion; resulting in poor turn-around time for the ships. In Case 2, as the Berth Spacing is increased from 200 m the results for the BCSP and STTP show a difference in the distance travelled and ship service time required to turn-around a ship. The distance travelled in Case 3 is 9226 km, a result of the quay length set at 1200 m and the Berth Spacing set to 300 m. This differs much from the BCSP, in that in the same case, 7542 km are travelled with an added ship service time of nearly 10 hours.

The general conclusion is that both distance and turn-around time is improved when finer berth spacing is used. A berth spacing of one meter is best to use when positioning an arriving container ship as opposed to berth spacing of 100 m or more, which is often used in the port industry. However, such an approach may require additional investment and increase the complexity of the organisation.

The resulting crane allocation varies between the two policies. The cranes that are assigned are different between the two policies as well as their positions. Not surprising as they are closest to the stacks, the middle cranes are assigned the most often, while the end cranes are working the least. During the process of loading and unloading, the final position for the cranes may indicate the amount of distance that they have travelled along the quay. In the STTP, the cranes assigned are more “balanced” along the quay than the BCSP. On the other hand the distances travelled are much further compared when using BCSP. The choice of policy is certain to influence the crane allocation and assignment.

7. Conclusion and Future Work

The results of the simulation experiments indicate that simulation as a backbone for a DSS can be useful for evaluating policies for berth allocation. The choice of policies has a strong influence on both the turn-around time and distances travelled by the SCs. The experiments indicate that the shorter length of the berth spacing the better. This suggests that a dynamic berth allocation is better than using fixed berths, which is a common practice. In addition, the results indicate that an informed choice of berth assignment policy can provide better use of the available resources, e.g., by reducing turn-around time and/or distance travelled by the SCs.

In order to compare the policies in monetary terms, costs could be introduced for SC distance travelled and to the Ship Service Time. The use of such costs may assist CT managers further in developing better routines and policies.

The BAMS tool is addressing only one subsystem, marine side, of a CT system. Future work would include extending it to model also the other subsystems, such as the yard or “land” side. The CT managers have explained, through personal interviews and an informal questionnaire that the configuration of container stacks in the yard can also determine the *turn-around time*.

The development of BAMS has provided much experience in attempting to model and simulate a system as complex as CT. The original strategy of modelling a whole CT system proved to be quite daunting, thus the strategy of modelling and simulating the sub-systems of the CT has proven to be more manageable. The BAMS is but one module that will be coupled with additional CT models, in developing a full DSS for CT managers.

8. Acknowledgements

This work has been partially funded by Karlshamn Municipality. The following port industry representatives have provided useful information necessary for the development of BAMS: Bengt Melin, Port of Karlshamn, Bill Miller, NIT; David Thomas, SGT; and Kjell Svensson, Port of Gothenburg. Appreciation for assisting in the software development: Anatoli Tcherviakov for the original software and Piotr Tomaszewski and Simon Kågström for assistance in debugging part of the code.

9. References

BRUZONE, A. G.; GIRIBONE, P.; REVETRIA, R. (1999), *Operative requirements and advances for the new generation simulators in multimodal container terminals*, Proceedings of the 1999 Winter Simulation Conference, Society for Computer Simulation International

FRANKEL, E. G. (1987), *Port Planning and Development*. John Wiley & Sons. New York, US

JEFFERY, K. (1999), *Recent Developments in Information Technology for Container Terminals*. Cargo Systems Report. IIR Publications. London, UK

KIA, M.; SHAYAN, E.; GHOTB, F. (1999), *The importance of information technology in port terminals operations*. International Journal of Physical Distribution & Logistics Management, 30/4, pp.331-344

LEATHRUM, J.; KARLBERG, L. (2000), *Analyzing the sensitivity of simulation parameters*, Proceedings of the Summer Computer Simulation Conference, SCSC July

NISHIMURA, E.; IMAI, A.; PAPADIMITRIOU, S. (2001), *Berth allocation planning in the public berth system by genetic algorithms*, European Journal of Operations Research 131, pp. 282-292

OJALA, L. (1992), *Modelling approaches in port planning and analysis*, Turku School of Economics and Business Administration. Turku, Finland

Appendix 1: Data Collected from Containerisation International and Author's own work

Ports	Geographic Area	Total TEU	Terminals	Total Quay length m	Number of Berths	Number of TEU per m of Quay Per Year
Hong Kong	East Asia	17 900 000	8	6 791	22	2 636
Singapore	South East Asia	15 520 000	5	6 453	44	2 405
Busan	North East Asia	8 072 814	10	11 040	62	731
Kaoshiung	East Asia	7 540 524	5	6 047	22	1 247
Shanghai	East Asia	6 340 000	3	2 281	11	2 779
Rotterdam	Northern Europe	6 102 000	13	12 375	20	493
Shenzhen	East Asia	5 076 435	3	5 600	21	907
Hamburg	Northern Europe	4 688 669	7	8 843	31	530
Long Beach	North America	4 462 958	9	7 806	38	572
Antwerp	Northern Europe	4 218 176	8	13 080	72	322
Bremen/Bremerhaven	Northern Europe	2 896 381	3	4 000	15	724
Gothenburg	Scandinavia/Baltic	697 000	2	3 065	8	227
Gdynia	Scandinavia/Baltic	217 024	1	978	5	222
Copenhagen-Malmö	Scandinavia/Baltic	129 000	2	1 300	6	99
Riga	Scandinavia/Baltic	101 023	1	445	3	227
Hamina	Scandinavia/Baltic	93 851	1	699	7	134
Rauma	Scandinavia/Baltic	83 850	1	2 542	16	82
Helsingborg	Scandinavia/Baltic	82 000	1	1 000	3	82
Lübeck	Scandinavia/Baltic	81 300	3	1 775	8	46
Klaipeda	Scandinavia/Baltic	51 675	2	840	4	62
Stockholm	Scandinavia/Baltic	35 637	1	480	1	74
Vasteras	Scandinavia/Baltic	30 400	1	540	6	56
Gdansk	Scandinavia/Baltic	20 476	1	275	1	74
Szczecin-Swinoujscie	Scandinavia/Baltic	19 960	1	473	1	27
Esbjerg	Scandinavia/Baltic	17 300	1	700	2	25
Wallhamn	Scandinavia/Baltic	15 782	1	709	4	22
Ghent	Northern Europe	15 590	1	560	4	31
Karlshamn	Scandinavia/Baltic	3 000	1	500	2	5
Rostock	Scandinavia/Baltic	1 450	1	743	4	2
Venstpils	Scandinavia/Baltic	256	1	454	2	1
Iskenderun	East Med/Black Sea	30	1	1 008	6	0

Appendix 2: Results of Simulation Experiment 1 and Simulation Experiment 2

		Straddle Carrier Distance							
		Berth Closest to the Stack policy (BCSP)				Shortest Turn-around Time policy (STTP)			
CASE 1: Best Scenario									
Berth Spacing		x 1 m	X 100m	x 200m	X 300m	x 1 m	x 100m	x 200m	x 300m
length (m)	400m	7 981 660	7 902 072	8 439 200	8 633 140	7 981 660	7 902 072	8 439 200	8 633 140
	600m	7 329 344	7 444 216	7 691 072	7 628 264	7 335 332	7 688 132	7 885 012	7 760 572
	800m	7 178 544	7 299 216	7 340 816	7 628 264	7 178 544	7 299 216	7 421 632	8 138 624
	1000m	7 178 544	7 299 216	7 340 816	7 628 264	7 178 544	7 299 216	7 421 632	8 138 624
	1200m	7 178 544	7 299 216	7 340 816	7 628 264	7 178 544	7 299 216	7 421 632	8 138 624
CASE 2: Avg. Scenario									
Berth Spacing		x 1 m	X 100m	x 200m	X 300m	x 1 m	x 100m	x 200m	x 300m
length (m)	400m	7 578 400	7 708 056	8 201 152	8 395 092	7 835 244	7 708 056	8 201 152	8 395 092
	600m	7 339 184	7 437 800	7 634 656	7 783 420	7 415 168	7 769 316	7 828 596	7 841 756
	800m	7 261 600	7 363 200	7 443 200	7 709 448	7 805 320	7 518 000	7 640 416	8 219 808
	1000m	7 261 600	7 363 200	7 443 200	7 709 448	7 805 320	7 942 832	7 640 416	8 397 928
	1200m	7 261 600	7 363 200	7 443 200	7 709 448	7 805 320	7 942 832	7 640 416	8 397 928
CASE 3: Worst Scenario									
Berth Spacing		x 1 m	x 100m	x 200m	X 300m	x 1 m	x 100m	x 200m	x 300m
length (m)	400m	8 145 132	8 419 092	9 022 140	9 448 400	8 655 992	8 419 092	9 022 140	9 448 400
	600m	7 364 340	7 489 320	7 837 328	7 541 284	7 939 272	8 376 672	8 428 352	8 575 832
	800m	7 088 012	7 207 120	7 229 120	7 541 284	7 548 894	7 816 236	8 004 572	8 393 016
	1000m	7 088 012	7 207 120	7 229 120	7 541 284	8 165 364	7 766 280	8 602 360	8 670 348
	1200m	7 088 012	7 207 120	7 229 120	7 541 284	8 165 364	9 020 984	8 268 080	9 226 240
Ship Service Time									
		Berth Closest to the Stack policy (BCSP)				Shortest Turn-around Time policy (STTP)			
CASE 1: Best Scenario									
Berth Spacing		x 1 m	x 100m	x 200m	X 300m	x 1 m	x 100m	x 200m	x 300m
length (m)	400m	44,34	43,9	46,88	47,86	44,34	43,9	46,88	47,96
	600m	40,74	42,7	43,79	43,1	40,71	41,32	42,72	42,34
	800m	39,88	40,54	41,22	45,2	39,88	40,54	40,77	42,34
	1000m	39,88	40,54	41,22	45,2	39,88	40,54	40,77	42,34
	1200m	39,88	40,54	41,22	45,2	39,88	40,54	40,77	42,34
CASE 2: Avg. Scenario									
Berth Spacing		x 1 m	x 100m	x 200m	X 300m	x 1 m	x 100m	x 200m	x 300m
length (m)	400m	43,52	42,84	45,56	46,63	42,1	42,82	45,56	46,63
	600m	41,19	43,16	43,48	43,56	40,68	41,32	42,41	43,6
	800m	43,35	41,75	42,43	45,66	40,34	40,89	42,4	42,83
	1000m	43,35	44,1	42,43	46,61	40,34	40,89	42,4	42,83
	1200m	43,35	44,1	42,43	46,61	40,34	40,89	42,4	42,83
CASE 3: Worst Scenario									
Berth Spacing		x 1 m	x 100m	x 200m	X 300m	x 1 m	x 100m	x 200m	x 300m
length (m)	400m	48,08	46,77	50,12	52,49	45,25	46,77	50,12	52,49
	600m	44,08	46,52	46,8	47,63	40,9	41,6	43,64	41,89
	800m	41,92	43,41	44,46	46,62	39,37	40,03	40,16	41,89
	1000m	45,35	43,13	47,78	48,15	39,37	40,03	40,16	41,89
	1200m	45,35	50,1	45,93	51,24	39,37	40,03	40,16	41,89

Introduction to BALTPORTS-IT: Applications of Simulation and IT-Solutions in the Baltic Port Areas

Eberhard Blümel, Fraunhofer Institute for Factory Operation and Automation (IFF),
Magdeburg / Germany, bluemel@iff.fhg.de

Leonid Novitski, IDC Information Technologies, Riga / Latvia, idc@balva.lv

Abstract

The political changes in Europe have resulted in a rapid alternation of transport flows and of transport operations structure in the Baltic Port Areas of Associated Candidate Countries (Poland, Lithuania, Latvia and Estonia). The strategic use of e-work, advanced IT and simulation solutions provides real advantages in the development of logistic transport systems associated with Free Port areas. This paper provides an overview of the main results obtained in the project IST-2001-33030 BALTPORTS-IT "Simulation and IT Solutions: Applications in the Baltic Port Areas of the Newly Associated States" funded by the 5th IST Programme of the European Commission.

1. Introduction

The BALTPORTS-IT project is aimed at promoting and supporting the dissemination of knowledge gained during the execution of the successfully completed EC funded projects such as AMCAI, *Bluemel et. al. (1997)*, DAMAC-HP, *Bluemel et. al. (2000)*, and SPHERE, *Schmidt et. al. (1997)*, as well as its industrial utilisation and transfer of technologies, simulation models and information systems.

The political and economic of the future years in Europe require a major effort of reorganisation of logistic strategies of the Associated Candidate Countries (ACC). So it is very important to integrate the ports of these countries in European logistic chains. Necessary as well is an improvement of the ports' competitiveness. Furthermore, Free Port Areas need to be developed in contrast to the now existing central control systems. Those Free Port Areas should be composed of port authorities, agencies, forwarders, trucking companies, stevedoring and insurance companies, customs authorities, banks, railway and warehousing companies. The infrastructure must be improved so that a fast and steady flow of goods can be guaranteed. In order to reach these goals, the partners want to re-design IT-processes and give simulation based decision support. The BALTPORTS-IT consortium consists of international partners from seven European countries, Fig.1. Scientists from universities and research institutes work together with experts from ports, maritime and IT-companies to develop methods and tools for the support of the reorganisation of Baltic Ports in the ACC.



Fig.1: Partners of the BALTPORTS-IT project

2. Objectives

The scientific objectives of the BALTPORTS-IT include:

- set-up of the Baltic sub-regional Competence Centre for promoting and supporting the distribution of research knowledge in the field of advanced IT solutions, logistics and simulation with maritime applications, Riga (Latvia);

- dissemination of research knowledge gained during the execution of the EC projects AMCAI, DAMAC-HP and SPHERE and regional projects in the field of IT-solutions and simulation of harbour managing;
- industrial customisation and exploitation of information and simulation systems by involving user groups;
- developing recommendations for the application of results and thus creating new market opportunities;
- creating opportunities for the training of specialists in maritime information systems design and port logistics by using web-based technologies and distance learning courses.

The results of the project include the industrial customisation of information and simulation systems in collaboration with user groups from the Baltic region in order to provide new approaches for:

- the non-monetary evaluation of general characteristics for port operations;
- the optimisation of logistic operations in container terminals;
- the optimisation of logistic processes in oil terminals;
- a methodology of marine information systems design.

3. Approach, solutions and technologies

The approach of the BALTPORTS-IT project consists of the following steps, Fig.2:

- customising the results of preceding projects (AMCAI, DAMAC-HP and SPHERE) for applications by generalising them;
- generalisation of the results of applications to find generalised methods for the redesign of IT-processes and simulation-based decision support;
- dissemination of the experiences made during the applications.

Customisation of simulation systems of container terminal operations and port processes as well as of maritime information systems obtained in the EC projects AMCAI, *Bluemel et al. (1997)*, and DAMAC-HP, *Bluemel et al. (2000)*, and the SPHERE-Project are among the main tasks of the project BALTPORTS-IT. Selected results are described in the following.

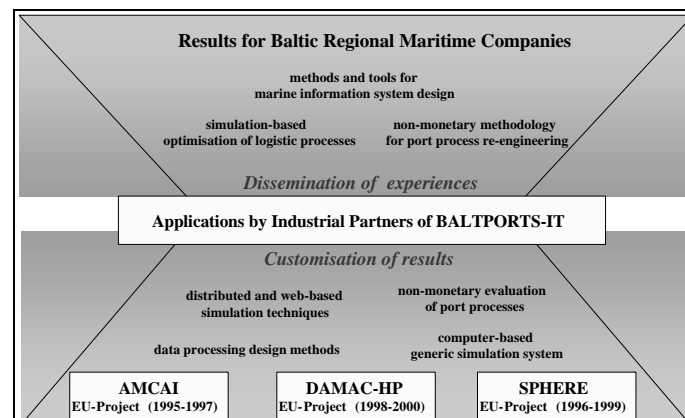


Fig.2: The approach of the BALTPORTS-IT project

4. Customisation of simulation and information systems

4.1. Application of simulation and modelling in Baltic port areas

Simulation was used for supporting managerial decisions at different levels of port operation planning and control, for instance, *Merkuryev et al. (2000)*:

- at the level of strategic planning, when designing a new terminal or redesigning an existing one: for the evaluation and comparison of different alternative decisions (e.g., related to planning terminal layout or updating terminal equipment);
- at the level of tactical planning, when optimising terminal operation by making decisions on management of resources (both labour and technical), taking into account schedules of vessels to be served during the considered period;

- at the level of operational planning and control, while planning the servicing of a specific vessel, taking into account cargo to be unloaded and loaded (e.g., berth planning, resource and yard allocation).

Simulation systems have been customised on the basis of the previous European and regional projects:

- the Baltic Container Terminal (BCT) Simulation System is based on a BCT simulation system that has been developed within the DAMAC-HP EU project;
- the Klaipeda Oil Terminal Simulation System is based on a simulation system that has been developed within a regional Lithuanian project;
- the Port Process Simulator is based on the SPHERE simulator that has been developed within the SPHERE EU project.

The considered simulation systems were customised in close co-operation with the port managerial staff in Riga (Baltic Container Terminal), Klaipeda (Klaipeda Port Authority and Klaipeda Oil Terminal) and Gdansk (Port of Gdansk and Gdansk Container Terminal), aiming to provide user-required characteristics of resulted simulation systems.

4.1.1. The Baltic Container Terminal Simulation System

The study concentrated on the logics of the basic technological and informational processes in the Baltic Container Terminal (BCT) serving operations. The modelling platform chosen was *Arena 5.0* of *Rockwell Software* since this software is well-suited for modelling statistically randomly distributed processes, which dominate in this model (e.g., vessel arrival schedule and the number of respective moves). This software also allows a good degree of dynamic visualization of process logics, which makes the model easy to comprehend and operate for users.

The logic of the BCT simulation system assumes a consequent rational detailed elaboration of processes, from general overview models down to more detailed ones with a developed hierarchical structure of embedded sub-models. The result of this approach, was the creation of a framework of four models, namely, *Merkuryev et al. (2002)*:

- *Model 1*: a general BCT model incorporated in the logical structure of agencies and other terminals (1st level of detailing);
- *Model 2*: a service processes model of every single vessel entering the port at up to three berths. This model portrays the logics of the simultaneous servicing of several vessels (up to three at a given time) and permits user changes in workload schedules as well as changes in the productivity of each individual resource;
- *Model 3*: a detailed model of every separate berth portraying in detail loading and unloading during every single move (container unit and re-stow containers) for a single vessel;
- *Model 4*: a detailed model of loading and unloading processes (included hatch covers) with underlying resource allocation and their monitoring with accuracy of up to 1 second.

Verification was performed using several traditional approaches including a walkthrough and involved independent as well as terminal specialists at both operational and managerial levels. Terminal management refined and approved the program logic, *Merkuryev et al. (2003)*. The model was calibrated by adjusting results from field measurement for time-lengths of the main technological operations to actually observed cycle times of terminal equipment. The model can be applied as a practical tool to assist management teams in the following tasks:

- real-time visualisation (monitoring) when changes in the BCT database are periodically transmitted to the model, which depicts the changes graphically;
- forecasting. Using the given input data, the model is run for k times, producing accumulated statistical averages and confidence intervals for future decision-making;
- statistical what-if analysis. A support tool for management to train personnel by simulating and analysing diverse work situations;
- adaptive control of vessel loading and unloading:
 - ? the model is run with the input data corresponding to the incoming vessel queried from the BCT database,
 - ? start/stop times of operations are recorded,
 - ? at determined points in time, the manager queries the database and compares current data with

forecasts,

- ? should there be significant deviations, the current data is used further in the model,
- ? the forecast parameters are determined,
- ? should the forecast be unsatisfactory, the resources in the model are changed and the model is run again with the current data from the database,
- ? this algorithm is followed until an acceptable solution for resource use is found.

The BCT Simulation System was developed by the team from Riga Technical University (V.Bardachenko, Y.Merkuryev and A.Solomennikov) and the BCT (F.Kamperman).

4.1.2. Distributed and Web-based simulation

Web-based simulation is an approach in the area of simulation that combines general Internet-based technologies with simulation tools and methodologies. The main focus in this approach is the forms of the general architecture and their applications in the framework of harbour models.

The basic web architecture is a client-server structure and specific architectures for web-based simulations are instances of this basic structure. The simulation software user is the client and interacts with different servers:

- Remote execution of existing simulation models – The client invokes a web browser, specifies input parameters of the simulation model in a special HTML document, submits this document to the server and starts the simulation machine on the server. Rather little model flexibility is provided for the user. Time consuming simulations can be executed on high-performance simulation servers;
- Local execution of downloaded simulation models – The server operates as an applet server. The complete simulation model is downloaded to the client site and is executed locally. The simulation program must be executable in the heterogeneous computer environment (e.g. simulation programs based on Java applet);
- Execution of a modifiable downloaded simulation model – This is an extension of the previous two examples. The server operates as a model repository or a file server that provides the simulation model source code . The client can modify the model. The model is executed either locally (if the appropriate simulation system is available) or remotely on an application server. This way, not only data but also the source code of the model is sent to the server;
- Download of input data – The client owns the simulation model and the server acts as a data server that offers special input data for the simulation (e.g. state of the real system).

Having described the general architecture forms, the following three application oriented architectures for harbour models may be introduced:

- *Simulation-based information system* - Harbour authorities provide an information system for their customers. Since not all customer queries can be answered based on analytical methods, a simulation-based information system is used. The heart of the system is a simulation model of the harbour processes Customers cannot change or modify the model;
- *Internal model adaptation* – Simulation models are used for the day-to-day operation of the harbour terminal. They are used to evaluate system behaviour with respect to its capacity for new business parameters or changes in operating conditions. This allows harbour management to analyse throughput and identify bottlenecks. Models may need to be adapted to the new conditions. When this cannot be achieved by changing parameters, model source code must to be changed;
- *Preparing external data* – Simulation models may be supplied with information stored on an external server (e.g. weather and tide forecasts). In such a case, the simulation model operates as a client obtaining the information from the server.

HLA-based simulation. The distributed simulation approach provides interoperability mechanisms during simulation runtime. One of the most recent standards in distributed simulation is the High Level Architecture (HLA).

In the case of HLA-based simulation, two categories of interoperability can be considered:

- *Internal interoperability* – The simulation model is divided into several sub-models (federates) that usually run on different machines. This allows the flexibility to configure the simulation for different purposes and levels of detail;

- *External interoperability* – In addition to simulation federates, an HLA federation can also have non-simulation federates that are not directly involved in the simulation.

Taking into consideration the aforementioned possible approaches to maintaining interoperability in the harbour simulation environment, some particular solutions were examined. These solutions are based on experience gained at Gdansk Container Terminal, a part of the Port of Gdansk. So far, the simulation has been used in the harbour operations based on stand-alone solutions. However, as pointed out above the limits of this approach need to be overcome in order to increase flexibility and applicability. The increase in interoperability is thus a major issue in future modelling and simulation applications in the harbour. Simulation models have to be integrated using some form of distributed information technology so they can be used not only by the different, sometimes geographically separated departments in the Seaport Authority, but also by other companies, business partners and customers operating in the harbour area. Web-based and HLA-based simulation approach was proposed and evaluated by the team from Fraunhofer Institute in Magdeburg (M.Schumann) and Otto-von Guericke University Magdeburg (B.Gebert, T.Schulze).

4.1.3. Customisation of marine information systems

The analysis of marine information systems designing methods and tools allows outlining the problems that must be solved when creating appropriate methodology, *Ginters et al. (1998)*:

- involving users (customers) without special knowledge in IT area in the design process;
- using company business and data models;
- application of mathematical and simulation modelling to formalise different stages of the design and customisation processes.

Creating business models based on a preliminary information survey of an organisation is a necessary phase of information system customisation, *Novitsky et al. (2001)*. Building flexible information systems requires the use of an enterprise model and a model-based development approach, in which the information systems are built, based on the requirements represented in the model.

The methodology of business process design consists of the following stages:

- Information survey of an enterprise using the Business Systems Planning (BSP) method, *IBM Corporation (1975)*
 - ? *Identifying the external environment*,
 - ? *Identifying the internal environment*. This phase involves defining all the factors that influence the environment within which a business must operate at the present time and in the foreseeable future,
 - ? *Business Planning*. The business plan defines objectives, competitive strategies, resource requirements and constraints for the organisation within the identified business environment,
 - ? *Business System Analysis (BSA)*. The BSA identifies the structure of an enterprise, the functional areas, the functions and sub functions both in the present and in the future. The information required by these functions is analysed. User views are one source of information describing the data needs;
- Creation of formal business process specifications.

LIS Technology, *Ginters et al. (1998)*, or Piece-Linear Aggregate (PLA) mathematical formalism, *Pranevicius (1992)*, are employed at this stage. The methodologies were used to customise different information systems, Table I.

Table I: Applications of BSP, LIS and PLA

Application \ Enterprise	Ventamonjaks	Klaipeda State Sea Port Authority	Klaipeda Oil Terminal	Insurance Company BALVA
Business process analysis and re-engineering	LIS	PLA	PLA	LIS, BSP
Simulation of oil terminal			PLA	
Incorporation of marine insurance IS into information management system				BSP, LIS

Customisation and application of marine information systems was done by the teams from Klaipeda University of Technology (KUT) and Klaipeda State Sea Port Authority (D.Makackas, V.Pilkauskas, H.Pranevicius, A.Zygus), from IDC IT and BALVA (L.Novitsky, V.Ragozin, M.Uhanova, E.Viktorova) and LIS and Ventamonjaks (E.Ginters, F.Rekners, A.Aumalis, L.Strujevics).

4.1.4. Other solutions

Among other solutions obtained within the frameworks of the BALTPORTS-IT project we can mention:

- Port Process Simulator from University of Ulster (F.-A.Schmidt, R.Yazdani). The idea behind the Port Process Simulator (PPS) is that a single software system in the form of a generic, customisable shell can simulate a complete port and its processes, i.e. infrastructure, suprastructure, aquastructure, equipment as well as the movements of ships and land vehicles. The simulator software can be applied to any port with the introduction of picture files of the port's approaches and layout. This is complemented by port-specific input data related to vessels, land vehicles and cargoes. The PPS has been applied to the ports of Gdansk, Klaipeda and Riga;
- The Klaipeda Oil Terminal Simulation System from KUT. The Simulation System of Klaipeda Oil Terminal evaluates two groups of characteristics. The first group of characteristics is used in queuing theory. Examples of such characteristics are occupation coefficients of embankments, platforms, reservoirs, etc. The other group of characteristics is used to analyse real-time systems, *Pranevicius and Makackas (2001)*:

- ? Probabilities that a ship that has arrived finds the volume of oil needed,
- ? Probability that the time a ship remains in a harbour is less than some limit value, etc.

To create the model of the Klaipeda harbour oil terminal, an aggregate approach was used to formalise terminal operation, *Pranevicius (1992)*. The ARENA simulation system was used to implement the simulation model software;

- A Non-Monetary Evaluation Methodology for Small and Medium-Sized Ports. The objective of the non-monetary evaluation methodology is to provide:
 - ? A management and planning tool providing expressions of "usefulness" and indications of "how" and "where" to improve the efficiency of the port under investigation in terms of: its operational capability, and / or, its institutional, regulatory, managerial and operational frameworks,

as regards local conditions and the expectations of the port community or the port authority involved.

The anticipated results are:

A generic methodology applicable to small and medium-sized ports supporting their operational planning and management by providing:

- ? Lists of port system components,
- ? Values describing the presence and operational performance of components,
- ? Ideal values or criteria for components,
- ? Preferences for components depending upon their contribution to the port system,
- ? Expressions of "usefulness" or utility,
- ? Indications of "where" and "how" improvements in the port efficiency can be achieved.

A monetary economic appraisal of a port can only take entities into account that can be expressed in monetary terms. It cannot cater for the rather disparate nature of institutional, regulatory, managerial or operational frameworks involved in the management and the operation of a port or a terminal. Nor can it take account of the access to, the layout of and the cargo handling facilities existing in a port or terminal or the flows of commodities, vehicles, personnel and information. However, multi-attribute utility techniques (MAUT) are ideally suited for the task at hand and assisted in the required development on a "Non-Monetary Evaluation Methodology".

During the project's workshops MAUT and its application were introduced by F.-A.Schmidt to personnel from the Port Authorities of Gdansk and Klaipeda and BCT to monitor the utility of the ports.

5. Setting up of the Baltic Sub-Regional Competence Centre in the field of logistics, advanced IT-solutions and simulation with maritime applications

5.1. Current state

The Baltic Sub-Regional Competence Centre (BSRCC) is a virtual structure aimed at bringing together industrial users, universities and research institutions around the common topic of, e.g. “Logistics, IT-solutions and Simulation with maritime applications”.

Who can benefit from the services?

- Specialists in freight transport and freight transport related logistics;
- Port managers and port consultants;
- Specialists in IT-solutions;
- Specialists from companies operating in port areas in e.g. freight forwarding, stevedoring, banks, agents, insurance, customs, students and academic staff.

What are the services offered?

- Consulting in transport, logistics, marine insurance and IT-solutions, e.g. analysis of cargo flows, business process analysis and re-engineering, improvement of IT-solutions and facilitating business partnerships;
- Realization of research projects according to the requirements of user groups;
- Education: lectures, seminars and computer-based distance learning and training;
- Distance training courseware in logistics information systems;
- Providing simulation models of harbour processes to achieve more transparency of processes and to discover the potential for optimisation;
- Establishing a network of excellence for validation, generalization, and dissemination of research knowledge.

What are the advantages of collaborating with the Competence Centre?

- The BALTPORTS-IT project's partners can provide customers with different state-of-the-art solutions within a short time;
- A network of excellence created around the Competence Centre guarantees the necessary resources and high level of research;
- Collaboration with Competence Centre will help customers to increase the profitability of their businesses.

Power Point Presentations of the following solutions are accessible at the BSRCC Web-site (www.balticIT.com):

1. Simulation System of Container Terminal;
2. Port Process Simulator;
3. Simulation System of Oil Terminal;
4. Combining Simulation and Information Systems;
5. HLA- and Web-based techniques;
6. Non-Monetary Evaluation Methodology for Ports;
7. LIS Technology;
8. Marine Insurance Information System;
9. Teaching and Training in Logistics Information Systems;
10. Intermodal Database.

5.2. Prospects

During one of our next projects, for instance, eLOGMAR-M, the BSRCC infrastructure will be further developed and branch offices in the other Baltic States will be established. Another possibility for strengthening the role and possibilities of BSRCC is to include it in the project coordinated by Fraunhofer IFF/FhG, which is now under preparation. This project is devoted to setting up Trans-European Centre in Logistics and BSRCC could be considered as one of regional centres of the whole distributed offices' network.

6. Acknowledgement

The presented activity was supported by the BALTPORTS-IT project “Simulation and IT-solutions: Applications in the Baltic Ports` Areas of the Newly Associated States” funded under the IST 5th Programme of the European Commission.

References

BLUEMEL, E. et. al.. (1997), *Managing and Controlling Growing Harbour Terminals: Application of Modern Concepts in the Automated Information Management in Harbours by Using Advanced IT-solutions*, SCSI, San Diego, Ghent, Erlangen, Budapest

BLUEMEL, E. et. al. (2000), *Simulation and Information Systems Design: Applications in Latvian Ports*, JUMI Ltd., Riga

GINTERS, E.; NOVICKY, L.; VIKTOROVA, E.; BLUEMEL, E. (1998), *Data Processing Systems Design*, Riga Technical University

IBM Corporation (1975), *Business System Planning. Information System Planning Guide*, New York: White Plains

MERKURYEV, Y.; TOLUJEV, J.; VISIPKOV, V.; MERKURYEVA, G.; KAMPERMAN, F. (2000), *Simulation-based analysis and management of logistics processes at the Baltic Container Terminal*, Proc. HMS 2000 Harbour, Maritime & Multimodal Logistics Modelling and Simulation, Portofino, Italy, pp.150-156

MERKURYEV, Y.; BARDACHENKO, V.; SOLOMENNNIKOV, A.; KAMPERMAN, F. (2002), *Multi-level simulation of logistics processes at the Baltic Container Terminal*, Proc. HMS2002 and MAS2002, International Workshops on Harbour, Maritime and Multimodal Logistics Modelling & Simulation, ed. A. G. Bruzzone, Y. Merkurjev, and R. Mosca, Genoa University, Genoa, Italy, pp. 78-83

MERKURYEV, Y.; BARDACHENKO, V.; KAMPERMAN, F.; SOLOMENNNIKOV, A. (2003), *Multi-level simulation of logistics processes at the Baltic Container Terminal: Calibration Study*, Proc. International Conference on Modelling and Simulation of Business Systems, Kaunas University of Technology, Kaunas, Lithuania, pp. 274-278

NOVITSKY, L.; GINTERS, E.; MERKURYEV, Y.; MERKURYEVA, G.; RAGOZIN, V. (2001), *Industrial Customisation of Maritime Simulation and Information Systems in the Latvian Port Areas*, ESS'2001 Simulation in Industry 2001: 13th European Simulation Symp., Marseille, France, pp.78-84

PRANEVICIUS, H. (1992), *Aggregate approach for specification, validation, simulation and implementation of computer network protocols*, Lecture Notes in Computer Science, No 502, Springer-Verlag, pp.433-477

PRANEVICIUS, H.; MAKACKAS, D. (2001). *The Use of Aggregate Approach for Formal Specification and Simulation of Real-Time System*, Database and Information Systems, Kluwer Academic Publishers, pp. 189–198

SCHMIDT, F.-A. et. al. (1997), *Report on the Development of the SPHERE SMP Operation Simulation Program*, Report on the EC under the SPHERE Project, December

Robot Applications in the Field of Shipbuilding

Peter WEISS, Cybernetix, Marseille/France peter.weiss@cybernetix.fr

Yann LE PAGE, Cybernetix, Marseille/France yann.lepage@cybernetix.fr

Frédéric SCHOM, Cybernetix, Marseille/France frederic.schom@cybernetix.fr

Alain FIDANI, Cybernetix, Marseille/France alain.fidani@cybernetix.fr

Abstract

Different to many industrial fields like automotive or food industry, few robotic applications can be found in the domain of ship construction. But work on shipyards includes many activities which are known to be hard and sometimes health damaging for the personnel working on site. This paper presents a first approach of the introduction of a robotic welding system in the area of ship assembly, a trend of high importance in order to keep ship industry competitive in Europe. Obstacles and drawbacks for such a systems that have to be overcome to make this development successful are illustrated. The paper conclude with an outlook on future applications for robots in this field.

1. Introduction: From mass to customised production

The application of robots has become standard in many fields of industrial production. The automobile and food industry are only two classic examples. Robots and automated systems have proven their advantage in those production processes due to their efficiency. The reason for the advantage of automated systems in production can be divided into three main aspects: economic factors, technological factors and ergonomic factors, Weiss *et al.* (2003). The economic factors are obvious: Process automation decreases the fabrication times and thus the costs of the units produced. Technological factors can be the precision or quality needed in production that can not be achieved by manual production means like for example in the fabrication and assembly of micro-technological devices. Ergonomic factors are the relief of the human worker of health damaging or tiring work.

The field of shipbuilding has nevertheless been spared by this trend of automation so far. One of the main reasons for this is the fact that ships are typically customised products that are fabricated in low quantities. Coenen *et al* (2003)

The introduction of automation in this field calls for off-the-shelf systems with following properties:

- Reduced programming times since the task differ from one ship to another.
- The systems must be transportable since the robots needs to be installed inside the ship.
- The direction of the shipyards and the operators on site must be convinced of introducing robots in the production process.

2. Welding Robots in customised production

Several projects can be named that address to the area of welding in customised production. The *WeldMate* of the Fraunhofer Institute for Manufacturing Engineering and Automation presents a versatile welding robot that can overtake tasks that are instructed by the human operator, Hagele (2003). The task is once taught to the robot which repeats those then automatically. This system presents a possible answer to the problem of long programming times which represents a principle obstacle in the fabrication of low number of pieces or sophisticated structures.

The *NOMAD* project also aims at the problem of long programming times with another solution. Peter *et al.* (2002). *NOMAD* develops a robotic system for the welding of large steel structures. The assembled structures are placed in a work cell which is equipped with autonomous robots to execute the welding. The position of the robot and the welding equipment to the work piece will be determined by the use of vision based sensor systems, touch sensing, laser-based vision and through-the-arc seam tracking. The geometric information is merged with the CAD data of the model, enabling the robot to find the different weld joints.

3. Robotic Welder in Shipyards: the DockWelder Project

These different projects on welding robots show the feasibility of automation in the fabrication of complex, low-volume production. The DockWelder Project resembles in its aims to these projects. The objective is to develop a flexible automated welding system for the intervention during the assembly phase of the ship construction. Fig.1 shows the different steps in the construction of a vessel. The first three steps, the raw material reception and preparation, the fabrication of the steel plates and profiles and the fabrication of the blocks already offers possible applications for automated systems. However, welding activities during the vessel erection are limited nowadays to manual work.

The DockWelder robot is intended to be used at this stage of the shipbuilding process when the ship sections are closed and ready to be welded. The system is a vehicle, moveable by the operator, that can be positioned inside the ship structure. Once positioned, it is able to execute the welding jobs automatically by finding its own position in the working area and by following the welding line.

The consortium working on this project consist of five partner organisations: APS (D) as co-ordinator, ODENSE STEEL SHIPYARD (DK) and FINCANTIERI (I) as potential end users, and the MÆRSK INSTITUTE OF PRODUCTION TECHNOLOGY (DK) and CYBERNETIX (F).

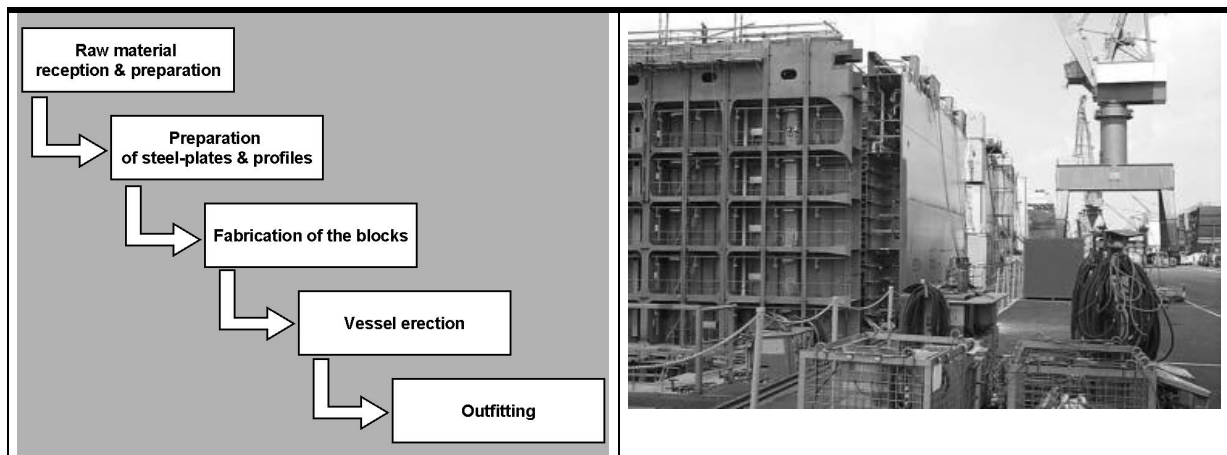


Fig.1: Fabrication steps in shipbuilding, *Andritsos and Perez-Prat (2000)*

Fig.2: Assembly of the ship hull.

Photo courtesy Odense Steel Shipyard

3.1. The System

The DockWelder system can be divided into three main components:

- The **Placer**, a vehicle which is used as the systems base. It is driven by an electric motor and steered by the operator.
- The **DT-VGT** (Double Tripod Variable Geometry Truss) which is a hydraulic manipulator that consist of two modules. With its length of approximately two meters it enlarges significantly the workspace of the robot.
- The **Welding Robot** is mounted on the end of the DT-VGT. It is a industrial robot of the company MOTOMAN which carries the welding equipment and sensors.

The aspect of the transport of the system to the working area and its installation is a key factor for the success of this project. While on one side it must be assured that the system (and its components like the power supply) can reach the inner parts of the ship without too much effort, it must later on, after the installation, be stable enough for the welding. For the transport to the working area, the concept of a container was chosen. It carries the welding source, control station, hydraulic pump and the vehicle itself. These containers are standard equipment on the shipyards and can thus be easily transported by a crane. Placed inside the ship, the DockWelder leaves its container, steered by the operator, and goes to the welding place. The vehicle is at this stage totally autonomous from any external energy source, since the motor is supplied by the batteries inside the vehicle.

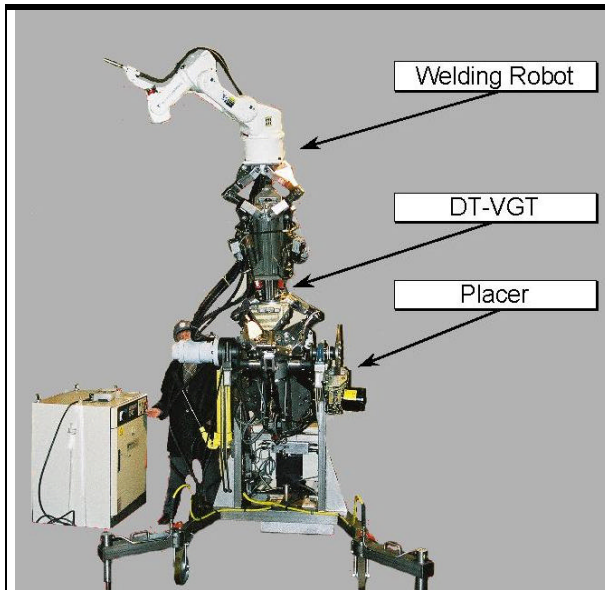


Fig.3: The main components of the DockWelder

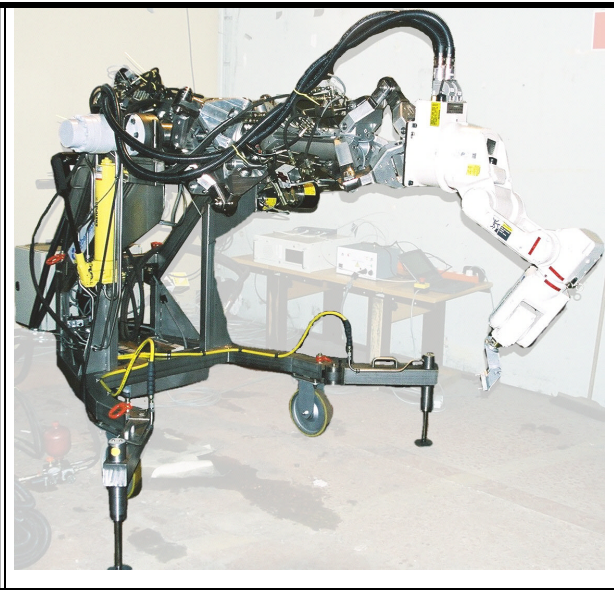


Fig.4: Side view of the system

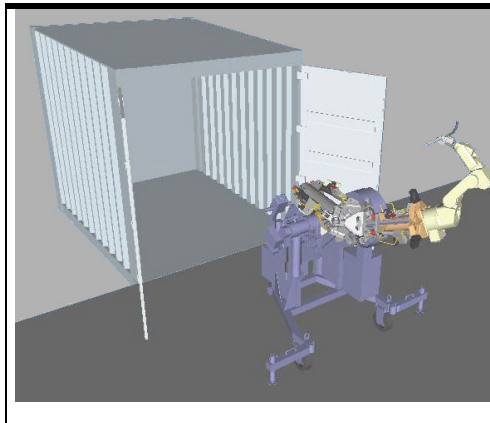


Fig.5: Transport container of DockWelder

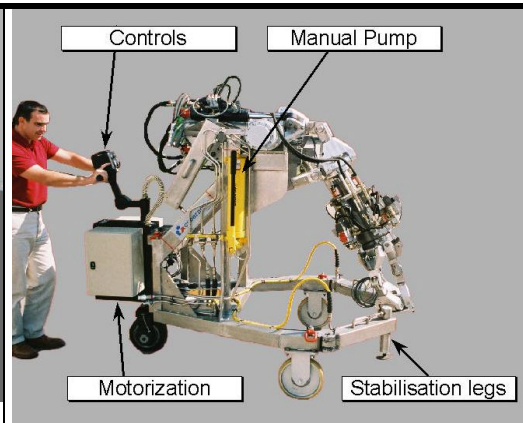


Fig.6: Motorization and stabilisation of the system

Arrived at the work place, the operator stabilises the robot by lowering the three legs to the ground. The legs are hydraulically actuated by a manual pump. At this stage all hydraulic and electric connections can be established.

Determining the robots position in the work area and finding the welding seam presents the next step in the operating procedure of the DockWelder. The system uses a laser scanner, that scans the room and calculates its position by comparing the acquired data with the CAD model of the section. A set of measurements is acquired for different positions of the laser spot, starting from 0° to 360° . The positions will be pre-programmed by setting the number n of acquisitions necessary. The result will be a set of points $[x_i, y_i, z_i]_{i=1 \rightarrow n}$.

A model derived from the CAD will be able to simulate a measurement for a given location and orientation of the laser, i.e. , simulating $z' = f(x, X_0, Y_0, \theta_0)$. The method is simple, it is just a computation of the intersections between the laser line(defined by X_0, Y_0, θ_0) and the different walls (defined by the CAD model).

A curve-fitting algorithm(non-linear least square problem) was used to minimise the difference of the two functions:

$$\sum_{i=1}^{N_{\text{points}}} r(q(i)) - r'(q(i), q_0, X_0, Y_0) = \text{Minimum} \quad (1)$$

The three variables X_0, Y_0, θ_0 are found iteratively, minimising the equation.

A min-max algorithm is tested at present. It is more efficient and stable. The idea is to compute the distance from each measured point to the closest item in the model, and minimise the sum of the distances by varying X_0, Y_0, θ_0 .

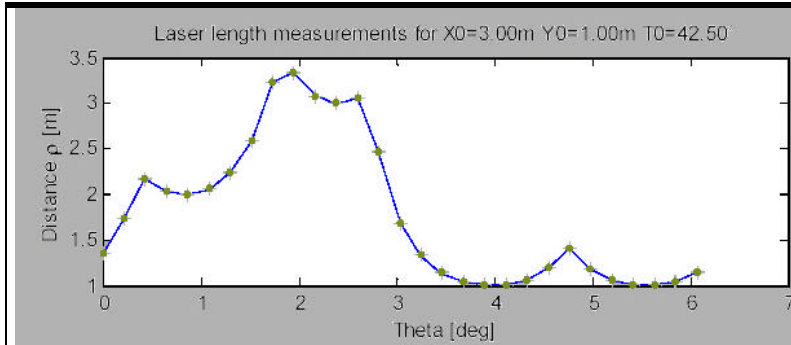


Fig.7: Output of the laser scanner

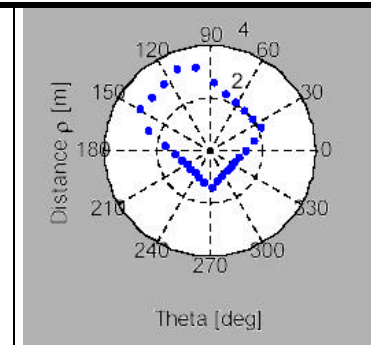


Fig.8: Output of the laser scanner

The output of the function will be the position of the system in the work section. This position is then delivered to the control system to calculate if necessary corrections in the trajectory of the robot. Figs. 7 and 8 show first results during the development of this device. A rectangular room is represented by the different dots. The vehicle's position is in the centre of the circle in Fig.8. For the case where the system is not able to calculate a position due to obstacles in the vision field, a control loop was introduced enabling the operator to exclude an angular field of the measurement.

The next step in the procedure is the deployment of the hydraulic DT-VGT in the welding position. The robot can weld on the floor of the section as well as the ceiling in approximately two meters height due to the large workspace of the DT-VGT. The DT-VGT is not moved during the welding procedure itself. The movement necessary for this is executed by the 6-axis Motoman SV3X robot which is mounted on the top of the DT-VGT. Both, DT-VGT and the Motoman robot are controlled by the same Man-Machine-Interface. The different welding tasks (and corresponding data like the welding time, etc) are pre-programmed by using the CAD data of this ship's section. The tasks need only to be executed by the operator to avoid long programming times.

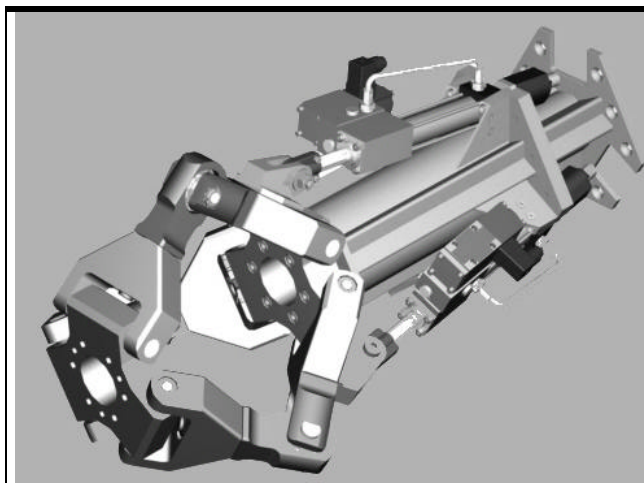


Fig.9: 3D Model of the DT-VGT.
Photo courtesy Odense Shipyards.

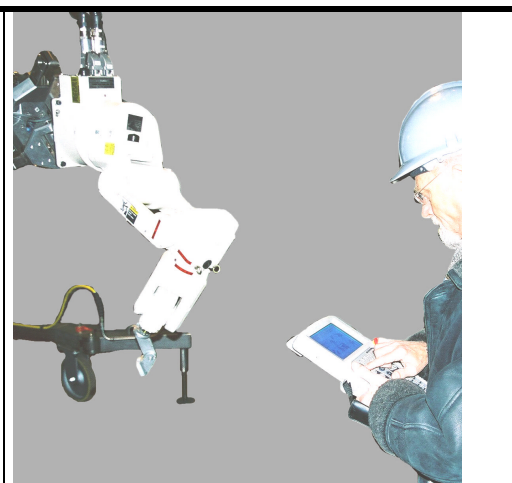


Fig.10: The MMI of the DockWelder will resemble those known today for industrial robots.

The precision of the process depends on the accurate position of the welding torch to the joint. The second position sensor of the system is the welding sensor which is an optical device detecting the gap between the welding parts. During the welding process the adaptive control mechanism is comparing the geometrical data of the joint area measured by the laser sensor with the parameter sets pre-calculated for this job. The acquired data is used to adapt the parameters of the welding robot and its integrated components like the welding torch and the wire feeder.

A key factor for the acceptance of the system by the personnel on site is the Man-Machine-Interface (MMI). It is clear that such a system –like any automated system- will need a certain degree of training before it can be used. But it must be assured that this system is easily understandable and that it is adapted for this working environment (a PC is typically not for example). Therefore a MMI in the form of a control pad was chosen and is under development.

The DockWelder project is at the moment of the edition of this paper in its integration phase. The coming months will be marked by intensive tests of the different components. The stability of the system was identified as critical aspect and the coming tests will show if the system meets the needs. These preliminary tests will be followed by tests in an authentic environment at the ODENSE and FINCANTIERI Steel Shipyards.

4. Conclusion

Robots will find their application in the field of shipbuilding as they have found in other industrial sectors before. Customised, low-volume production calls for flexible systems able to be easily adaptable to the different tasks. This paper presented an approach in the field of automatic welding during the ship assembly. While still many obstacles have to be surmounted, this system will show the feasibility to use robotic welding systems on shipyards.

Acknowledgements

The authors of this paper want to thank all partner organisations of the DockWelder project, namely APS (D), FINCANTIERI (I), MIP (DK) and ODENSE STEEL SHIPYARD (DK). The DockWelder project is financed by the European Commission under the GROWTH programme. We want to express here our thankfulness to the EC for making this research possible. The authors profited for the edition of this paper from information of and discussions with Mr. Fivos ANDRITSOS of the JRC and Mr. Keith HERMANN of Caterpillar. We want to thank both very much for this dialog.

References

- WEISS, P.; SCHOM F.; CHARDARD Y. (2003) *Intervention of robots in shipbuilding and maintenance*, 2nd Int. EuroConference on Computer and IT Applications in the Maritime Industries, COMPIT'03, Hamburg, pp.425-431
- COENEN, J.; NIENHUIS, U.; RIJKE, R.; VAN DER WAGT, J.; (2003) *Economic feasibility prediction for welding robots in shipbuilding*, 2nd Int. EuroConference on Computer and IT Applications in the Maritime Industries, COMPIT'03, Hamburg, pp.490-501
- ANDRITSOS, F.; PEREZ-PRAT, J. (2000), *The Automation and Integration of Production Processes in Shipbuilding*; Joint Research Centre of the European Commission, Report EUR 19663 EN
- HAGELE, M. (2003) Presentation at the CLAWAR conference, Karlsruhe November 2003
- PETERS, C.; HERMAN, K.; LYLINOJA, A.; (2002) *Manufacture of Large Steel Fabrications using Autonomous Robots*, 33rd Int. Symp. Robotics, ISR2002, Stockholm

An intelligent GA based AIS FATDMA scheduler

Vedeesh Sahajpal

ControlIT, Sadar/India 440001, vedeesh.sahajpal@norcontrolit.com

Abstract

A Multichromosome Genetic Algorithm is proposed to prepare the FATDMA schedule for an AIS base station for its periodic broadcasts. Termination criterion is made sharp by incorporating the existence theorem of an approximate but exact method, whereas fitness function is made more discerning by utilizing the lower and upper bound of this approximation method. The exactness refers to the tree based deterministic scheduling algorithm described elsewhere and referred in this paper, with proven optimality bounds for the solution obtained. Our method on the other hand is heuristics driven, a GA based approximation method, with an element of determinism thrown in as contained in the referred algorithm – this being the explanation of the intelligence part of the essentially blind Genetic Algorithm.

1.Introduction

An AIS base station needs to be programmed to schedule transmission of various AIS messages like 4,17,20 and 22 containing diverse vital information like UTC time, location of base station, DGNSS corrections and the base station's own reserved FATDMA slots and other exigency based broadcasts etc. A periodic scheduler is thus called for, to schedule transmission of the above-mentioned periodic messages whose expected transmission frequencies under various conditions are stipulated in [6].

Some other examples of perfectly periodic schedules arise from the wireless communication field. For example a Bluetooth device in sniff mode is obliged to listen to the network only in time windows defined in strictly periodic fashion. Another example is the concept of broadcast disk [8] where a server continuously broadcasts a database. If the broadcast schedule is perfectly periodic, it is extremely easy for a client to compute when will be the next occurrence of its desired item. In a perfectly periodic schedule it is also known how to interleave "index messages" between the data messages so that a randomly arriving client does not need to continuously listen until its desired data message is transmitted.

2. AIS communication essentials

The AIS is based on digital radio communication using Time Division Multiple Access (TDMA). The data link is divided into frames, each frame being 60 seconds.

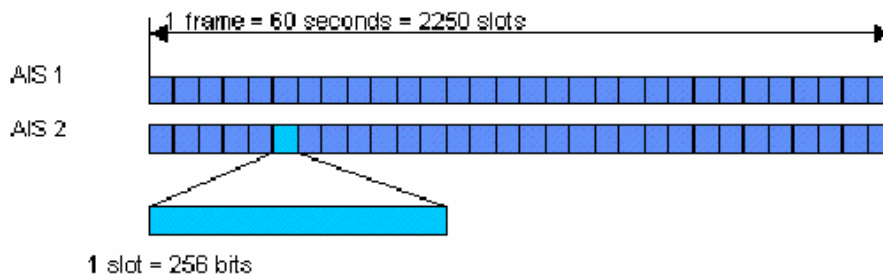


Fig. 1

The frame is further divided into 2250 slots. Each slot can hold a position report. The AIS operates in parallel on two frequencies within the maritime mobile band. These frequencies default to AIS1 and AIS2 (161.975 MHz and 162.025 MHz).

Four different types of protocols (RATDMA, FATDMA, SOTDMA and ITDMA) are used to send a message. Every ship is using the two designated radio frequencies to organize transmission and reception of messages in an autonomous self-organizing mode.

When ship A sends a message, she also assigns a future time slot to send the next message in. Other ships do the same and avoid the already assigned time-slots. There is a risk that two ships that are out of radio range of each other will use the same timeslot. If this happens, the one that is closest will be heard according to the modulation technique that is used. This forms a cellular system where every ship is her own center. This makes the system very secure even if it is overloaded with several hundred percent. The update rates of ships position reports are automatically adapted to current speed and maneuver; in a range between minutes down to two seconds, Table I.

Table I: Information update rates for autonomous mode

<u>Navigational Status</u>	<u>Reporting interval</u>
Ship at anchor or moored and not moving faster than 3 knots	3 min
Ship at anchor or moored and moving faster than 3 knots	10 s
Ship with a speed between 0-14 knots	10 s
Ship with a speed between 0-14 knots and changing course	3 1/3 s
Ship with a speed between 14-23 knots	6 s
Ship with a speed between 14-23 knots and changing course	2 s
Ship with a speed greater than 23 knots	2 s
Ship with a speed greater than 23 knots and changing course	2 s

The standard AIS messages consist of maximum 256 bits and are transferred with a bit rate at 9600 b/s. This ensures 2250 available time-slots within a 60 second period. With the highest report rate at 2 seconds interval, the system is limited by a theoretical limit of 75 ships. However, this is highly unlikely because of the very nature of ships this system is designated for; massive ships with low turn rate and speed.

2.1 Essential AIS FATDMA Messages

Since, in this paper we are focusing on FATDMA as applied within the AIS domain, let us take a closer look at the overall messages summary and the link access methods. For a summary of various AIS messages from 1 to 22, and the recommended link access scheme (FATDMA, RATDMA, SOTDMA etc.), please see [6]. For messages specific to FATDMA, let us take a little detailed look by quoting verbatim from [6] and [7] The idea is to put in one place all the diverse information applicable to us in this paper's framework and also to draw some motivation for the need for FATDMA access for these specific messages as felt by this author. Reader, who is more interested in the generic periodic scheduling problem without being bothered about the AIS FATDMA specific motivation, can skip this section.

Message 4: See: section 15.3 page 40 of [7]

If configured, the AIS Base Station should periodically generate the Base Station Report (Message 4), which is transmitted for reporting UTC time, date and position. The base station should normally transmit this message with a minimum reporting rate of 10 seconds. The base station should operate in this state

until it detects one or more stations that are synchronizing to the base station. It should then increase its update rate of Message 4 to 3 1/3 second (nominal) intervals, interleaved between both AIS channels. It should remain in this state until no stations have indicated synchronizing to the base station for the last 3 minutes. FATDMA reserved slots are used for these transmissions. SOTDMA communication state operation stays as defined in ITU-R 1371-1, A2, 3.3.7.2.2, except when slot time out is 0, then slot off-set equals 2250. The Configuration Message (CBM) should be used to configure the msg 4 transmission schedule.

Message 17: See Section 3.3.8.2.13 page 58 of [6].

This message should be transmitted by a base station, which is connected to a DGNSS reference source, and configured to provide DGNSS data to receiving stations. The contents of the data should be in accordance with Recommendation ITU-R M.823, excluding preamble and parity formatting.

See Annex B of [7] for following:

The reference receiver generates corrections with a high update rate. Normally a complete set of corrections is available each second. Dependent on the used communication link the data will be available at the AIS base station with a latency of 1 to 6 seconds. As described in chapter 3 the AIS base station will transmit the AIS VDL Message 17 within 3 time slots with a repetition interval of = 5 seconds. To fulfil the demands of integrity and accuracy it is important that the AIS base station will always use the most topical set of correction data that is available at that time.

See Section 3.3.8.2.16 of [6]

Message 20: This message should be used by base station(s) to pre-announce the fixed allocation schedule (FATDMA) for one or more base station(s) and it should be repeated as often as required. This way the system can provide a high level of integrity for base station(s). This is especially important in regions where several base stations are located adjacent to each other and mobile station(s) move between these different regions. Mobile stations cannot autonomously allocate these reserved slots.

The mobile station should then reserve the slots for transmission by the base station(s) until time-out occurs. The base station should refresh the time-out value with each transmission of Message 20 in order to allow mobile stations to terminate their reservation for the use of the slots by the base stations (refer to § 3.3.1.2).

Message 22: See section 3.3.8.2.18 of [6]

This message should be transmitted by a base station (as a broadcast message) to command the VHF data link parameters for the geographical area designated in this message. The geographical area designated by this message should be as defined in § 4.1. Alternatively, this message may be used by a base station (as an addressed message) to command individual AIS mobile stations to adopt the specified VHF data link parameters. When interrogated and no channel management performed by the interrogated base station, the not available and/or international default settings should be transmitted (see § 4.1).

One may thus regard these 4 messages as constituting the management aspects of the AIS protocol. FATDMA periodic scheduling thus fits well for the following reasons:

1. Periodicity on the time line, permits bandwidth saving. An AIS client, mostly a ship in this context, would thus seek these management messages only in these easily computable pre-designated time windows. The other time slots (the free band width) would be utilized for other dynamic and non-periodic messaging needs (RATDMA, SOTDMA etc.)
2. Temporal consistency is easily achieved when you know the exact time you would be receiving update information, for example, DGNSS corrections (Message 17) and also UTC time and position information (Message 4). Thus barring the latency inherent in DGNSS data collection [7 page 131], further delay is minimized. This aspect is brought out more clearly, with regard to the requirement of updating the base station reporting- rate when it identifies a station as trying to synchronize to it.

Section 3.3.8.1 of [6] gives the message summary of all the AIS messages. Remember all the time that FATDMA is reserved for Base Stations; see section 3.3.4.3 of [6]. Also note that the base station rate should increase to once per $3 \frac{1}{3}$ s after the station detects that one or more stations are synchronizing to the base station (see § 3.1.3.3.1, Annex 2, page 4 of [6]). The link access scheme (FATDMA, SOTDMA, RATDMA, ITDMA) for the various messages has been denoted in this summary. Two things need to be noted—either of the above 4 management messages could also access the link using a scheme other than FATDMA. Also, one of the following messages, other than the mentioned 4 management messages, viz, 6,7,8,10,12,13,14,15,16 and 21 could be aired on one of the FATDMA allocated slots.

The above message summary is not meant to give a comprehensive coverage of all the subtleties and rules contained in AIS protocol as described in greater details in quoted authoritative references. The summary is meant to motivate the need for FATDMA, and to see AIS FATDMA as yet another instance of the generic periodic scheduling problem as described below in problem statement section 4.

3. Need for heuristic method

It should be noted that perfect periodic schedules are not always feasible and that it is NP-hard even to decide whether a given set of requests admits a perfectly periodic schedule [4]] Thus one has to take recourse to heuristics and blind search methods like those based on Genetic Algorithms. Some recent papers [1] have proposed approximation algorithms based on minimizing the approximation ratio, with approximation ratio being the proportion between the requested period and the granted period. Existence theorems have been developed and also lower and upper bounds on the approximation factor under various measures have been proved in these papers.

Perfectly periodic problems can either be for the unit-length class of problems where each job is one slot long or the multi length class of problems where each job may have a different length. In the AIS protocol, different messages have differing sizes and hence occupy different number of slots when aired. Whereas messages—4, 20 and 22 are each one slot wide only, message 17 in particular, with its maximum possible size of 816 bits necessitates usage of up to 4 FATDMA slots. This has implications on the design of the periodic scheduler. This is thus a multi-length model problem (see further in the section 4 on problem statement and notation).

3.1 Our proposal with a Genetic Algorithm

GAs are essentially blind, wading through the solution space modifying it over iterations using crossover and mutation and constantly seeking a yet better solution under the watchful control of the fitness function. This paper achieves the following goals:

1. Proposes and implements a Multichromosome GA based algorithm to find a FATDMA periodic schedule with a fitness function essentially monitoring approximation ratio as a guiding criterion.

A multi-chromosome approach facilitates separation of independent features. Custom crossover and mutation operators can now more easily be designed due this clear separation between distinct attributes of the problem being solved.

2. Modifying the above by incorporating the existence theorem [1] to ensure that GA keeps searching as long as a guarantee for the existence of a solution is provided by the existence theorem, thus having a very refined termination criterion. It might be noted that GA are traditionally used in solving problems, which do not admit a direct closed form solution. Most often the terminating criterion is either an upper bound on the number of iteration or fitness function value exceeding a certain threshold. However, most problems necessitating the usage of GA do not suggest a way to calculate the threshold value. It is here where we use the existence theorem for another method, as a bait to keep pushing the GA towards an eventual better solution and not terminating mid-way frustrated after not finding a solution in certain number of iterations.
3. Incorporating the upper bound calculation on the approximation factor in the fitness function to ensure that the schedule produced by GA is at least as good or better than the published results for the best-known lower bounds by direct approximation methods (not blind search) as published by other authors.
4. An additional aspect of base station programming is to ensure that two neighboring base stations (say within 100 nautical miles, might be different countries) do not have conflicting FATDMA slots. Implications to the algorithm modification are nominal and we will present hints as to how cooperating neighboring competent authorities can utilize a client-server or a web-based backbone model to ensure that the FATDMA reservations conflicts between geographically close base stations is eliminated

4. Problem statement and notation

Instances. An instance of the perfectly periodic scheduling problem is the set of n jobs (called reservations in AIS parlance), $J = \{i = (l_i; p_i)\}_{i=1 \dots n}$, where l_i is the size or length of job i , and p_i is the requested period of job i . The maximal length of a job in an instance J is defined by $B_J = \max\{l_i | i \in J\}$. The maximal and minimal values of the requested periods in instance J are defined by $T_J = \max\{p_i | i \in J\}$ and $t_J = \min\{p_i | i \in J\}$. The ratio $R_J = B_J/t_J$ is called the extent of J . For the single server model we assume that $R_J < 1$ (otherwise no schedule can satisfy the request of J). We omit subscripts when they are clear from the context.

Schedules. A schedule S for an instance J is a set of infinite sequences of start times $S = \{I_1, \dots, I_1\}$, such that $I_i = \{A_{i1}, A_{i2}, A_{i3}, \dots\}$ where A_{ik} is the non-negative integer for each i, k . We say that job i is scheduled in time slot t if for some k , $A_{ik} = t < A_{ik} + l_i$. A schedule S is said to be *perfectly periodic* if for each job i there exists a granted period p_i such that for all J , $A_{i,j+1} - A_{ij} = p_i$. The generated periods may be different from the requested periods, but the job lengths cannot be truncated by the schedule. Periodic schedules can be represented by their cycles, i.e. a sequence of time slots whose length is the least common multiple of job periods.

The *requested bandwidth* of job i is defined by $\beta_i = l_i/p_i$. The total bandwidth of an instance J is defined by $\beta_J = \sum_{i=1 \dots n} \beta_i$. If $\beta_J > 1$ there is no feasible schedule for the instance. The *free bandwidth* of an instance J is defined by $\Delta_J = 1 - \beta_J$

Quality measures. Let J be the instance for perfectly periodic scheduling problem and let S be the schedule for J . The *individual ratio* of job i in S is defined by $\rho_i^s = p_i^s / p_i$. The MAX measure is simply the maximum over the individual ratios, defined formally by $C_{MAX}(J, S) = \max \{ \rho_i^s | i \in J \}$.

The AVE measure is the weighted average of the individual ratios, where weight of i is its requested bandwidth.

$$C_{AVE}(J, S) = (\sum_{i=1 \dots n} \beta_i \rho_i^s) / \beta_J = (\sum_{i=1 \dots n} l_i p_i^s / p_i^2) / \beta_J$$

5. Genetic Algorithms

We give a highly condensed summary of Genetic Algorithms here, as it is now a widely studied field. The ingenuity in the usage of GA lies in the construction of crossover and mutation operators, and we spend more time explaining our constructions. Liberal use of metaphor has been used to reinforce the evolution of this line of algorithms from natural genetics and from the principle of survival of the fittest. In particular, the two theorems borrowed from [1] have been interpreted in this manner, to weave them seamlessly in the heuristics fold. This combination of deterministic and heuristic principles in a common setting is the major idea of this paper and thus we feel that this extra emphasis on this seamless integration is worth the effort.

Genetic algorithm is a guided random search algorithm based on the principles of evolution and natural genetics. It combines the exploitation of past results with the exploration of new areas of the search space. By using *survival of the fittest* techniques and a structured yet randomized information exchange, genetic algorithm can mimic some of the innovative flair of human search. Genetic algorithm is randomized but not simple random walks. It exploits historical information efficiently to speculate on new search points with expected improvement. Genetic algorithm maintains a *population* of candidate solutions that evolves over time and ultimately converges. Individuals in the population are represented with *chromosomes*. Each individual has a numeric *fitness* value that measures how well this solution solves the problem. Genetic algorithm contains three operators. The *selection* operator selects the fittest individuals of the current population to serve as parents of the next generation. The *crossover* operator chooses randomly a pair of individuals and exchanges some part of the information. The *mutation* operator takes an individual randomly and alters it. As natural genetics, the probability of applying mutation is very low while that of crossover is usually high. The population evolves iteratively (in the genetic algorithm terminology, through *generations*) in order to improve the fitness of its individuals. The structure of genetic algorithm is a loop composed of a selection followed by a sequence of crossovers and mutations. Probabilities of crossover and mutation are constants and fixed in the beginning. Finally, genetic algorithm is executed until some termination condition is achieved, such as the number of iterations, execution time, results stability, etc. It is here that we have refined the termination criterion based on the theorems proved in [1].

6. A straightforward multi-chromosome based periodic scheduler

A tricky question is how to represent a schedule in a way suitable for a heuristic algorithm. We decide on the representation in Fig.2. This is thus a straightforward mapping from the problem description to the GA algorithmic settings. The (p_i, m_i) binding prescribes the bandwidth allocation generically and plays a role in quality factor analysis, whereas the essential slot conflict resolution using Chinese Remainder Theorem works on the (s_i, p_i) binding; A value of m_i not equal to 1, makes it a multiple length problem. In particular, message 17 of AIS with a packet size extending up to 816 bits (thus 4 slots), makes it a multiple length problem.

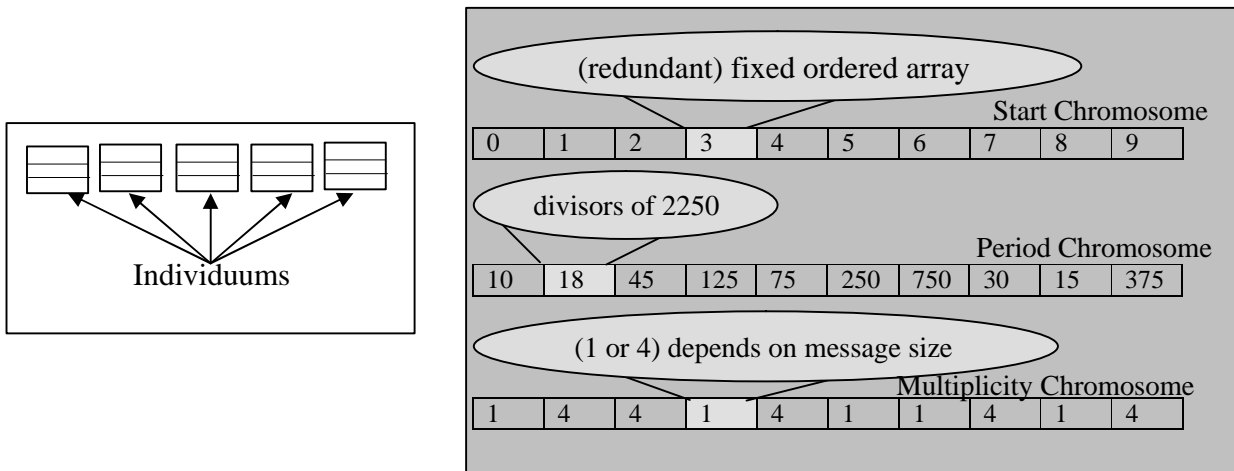


Fig.2: Structure of a multi-chromosome GA

6.1 Crossover

The crossover operator in a multi-chromosome setting could be implemented in either of the following two ways:

- Only a single chromosome is used for crossover, or
- All chromosomes of an individuuum undergo the crossover process.

In the first case a single chromosome of each individuuum is randomly chosen, and the crossover operator is applied. In the second case, the standard operator is

We have decided not to subject the chromosome tagging the start points to undergo crossover in our composite crossover operation. This feature makes it easier to design the check and repair algorithm, which is the most complex part in genetic algorithm design. One could think of fixing the start chromosome, thus maintaining order, and obviating the need for an extensive repair step. This would thus essentially make the start chromosome redundant, in the programming sense. An extra index array would however be required to keep track of the good reservations as against the bad ones. This maps more intuitively with the generic slot line. You move reservations around (crossover), change (mutate) an occasional period gene, to bring in the lost diversity, and mate forward in successive populations hoping for a better reservation approximating the desired pattern after eliminating the conflicts (fitness function job to correct this).

We would thus only be talking about the period and multiplicity to be matched with the fixed starting points on the slot line. A good-match gene tuple (start, increment, number of slots) would give a higher score to this particular gene tuple in the composite (multi-chromosome) genome, thereby raising the overall score of the composite genome.

Our solution is not thus the full chromosome, but rather only select genes of it. The redundant genes could still constitute valid reservations, but more likely they would not. As the position of the better reservations (the start slot) could keep shifting in the best chromosome from iteration to iteration, the fitness function has to be alert in utilizing and updating with respect to the positions of the best reservations from iteration to iteration.

6.2 Mutation

A random number out of the valid gene types for the period and multiplicity chromosomes are generated, guaranteeing a valid gene after mutation.

6.3 Initial Population

Each individual of the initial population is a clone of the input request individual. The period genes are spread over divisors of the frame size, namely 2250. The distribution of these divisors is uniform over the various individuals in a population because of random choice from the set of all divisors. The length chromosome is slightly biased to have more entries corresponding to the unit length messages (all except message 17 are unit length in AIS protocol).

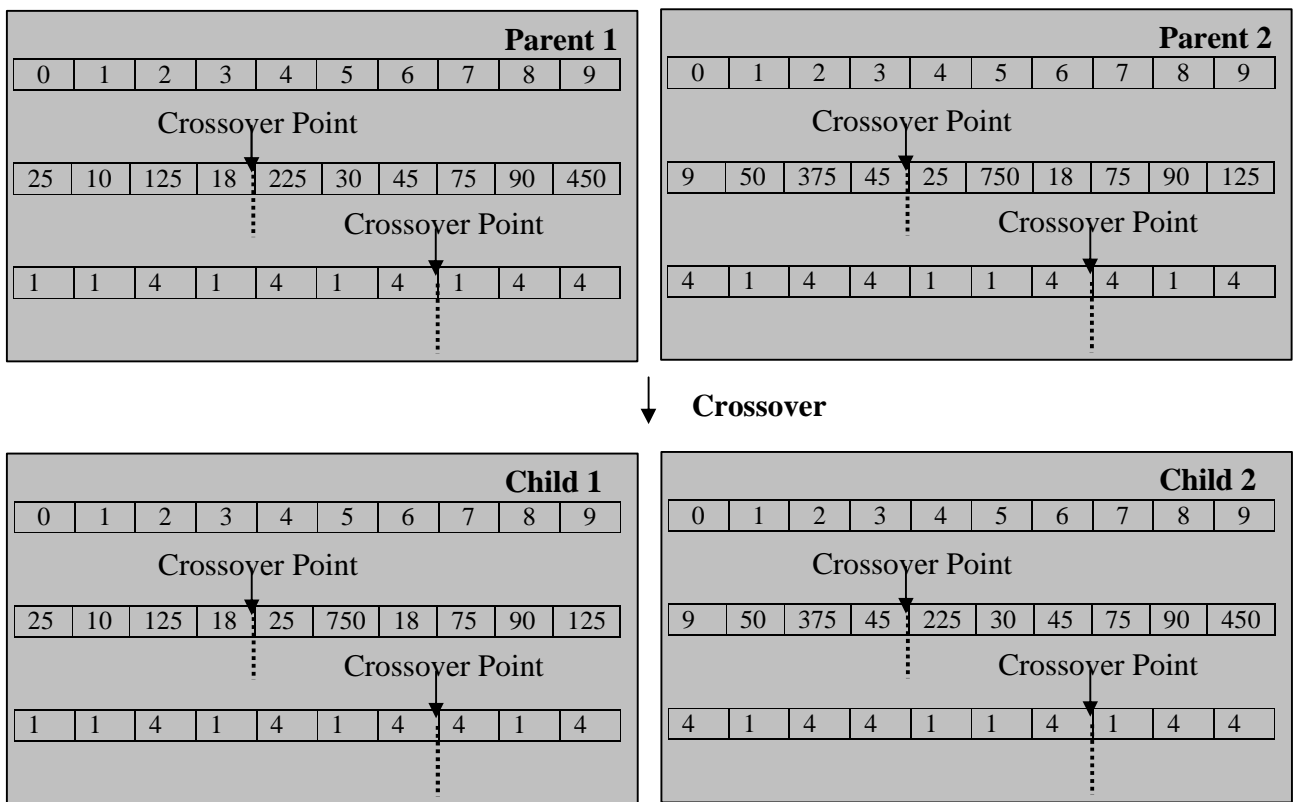


Fig.3: Crossover Example

6.4 Fitness function

Let us start by quoting the well-known Chinese remainder theorem. Essentially it gives an easy gcd based method to look ahead into a pair of period sequences, and inform us whether the two sequences would have conflicting entries at any location in their independent runs. In simple words, it says that if the gcd of the two periods divides the difference of the starting points of the two sequences, then there would be a point of conflict between the two sequences. Formally:

Theorem 1: Chinese Remainder Theorem

Two simultaneous congruences $n = n_1 \pmod{m_1}$ and $n = n_2 \pmod{m_2}$ are only solvable when $n_1 = n_2 \pmod{\gcd(m_1, m_2)}$. The solution is unique modulo $\text{lcm}(m_1, m_2)$.

When m_1 and m_2 are co-prime their gcd is 1. By convention, $a = b \pmod{1}$ holds for any a and b .

Chinese remainder theorem would essentially identify conflict between two periodic sequences only. But Chinese remainder theorem cannot do better than informing us about the conflicting reservations. It does not suggest a way to eliminate the conflict. Conflict removal is the job of crossover and mutation operators.

Thus our naïve fitness function consists of the following bare minimum checks and associated rewards and penalties being assigned to the genome under review:

1. Use Chinese remainder theorem to identify a conflict. For each conflict of a reservation (a triple) with any other reservation, add a penalty point. Thus reservations with maximum conflicts will accrue more penalties. Note that this is a symmetric operation and thus the penalty score will get doubled. It thus needs to be normalized by division by a factor of 2. The score of a chromosome consists of the individual scores of each of its constituting genes. Chromosomes with lesser internal conflicts will tend to score better than chromosome with higher internal conflict.
2. This second component of our fitness function is driven by the needs for a good quality solution. A straightforward quality factor is as per the discussion contained in section 4.

Please note that AIS specific detailed rules like channel switching etc., are not being incorporated in this presentation and thus not a part of the fitness function being discussed here.

7. An intelligent multichromosome periodic scheduler

GA algorithms encapsulate problem specific knowledge primarily in the fitness function. A well-written fitness function would sharply differentiate between a chromosome better approximating a solution than the one far from an ideal solution. Knowledge of what constitutes a good chromosome is indeed a problem specific attribute. Fitness function thus embodies a measure or metric of comparison between different chromosomes in the population. And the best metric is naturally proposed by the underlying problem being solved.

The class of problems seeking a GA approach does not always have the luxury of availability of any approximation algorithms. Where available such approximation algorithms can yield insights to construct better fitness functions.

The following paper [1] constructs a series of algorithms to approximate solutions to the perfectly periodic problem as described in section 4. The paper starts with an innocuous theorem establishing the existence of a solution and its quality. It goes on to refine the results to greater extents, culminating in the last theorem establishing the existence of a solution with a known upper bound on quality factor.

We are quoting two theorems from [1].

Theorem 2: For any given integers $B < t$, there exists an instance J with maximal job size B and minimal requested period t such that for all schedules S for J , we have

$$C_{AVE}(J, S) > 1 + R_J - \frac{2 + R_J}{t} = 1 + R_J - \frac{3}{t}$$

7.1 Adaptation of Theorem 2 in our GA framework

This first theorem above tells us that we cannot do better than the lower bounds proved. Thus genomes with a fitness values worse (higher) than this lower bound have a higher right to stay in population, whereas those with better (lower—hence a tighter ratio between granted and requested bandwidth) scores are immediate candidates for a raised eyebrow. Could this be a case of deletion and replacement from the population?. Read the paragraph below, for our reasoning against such complete mapping of deterministic principles on essentially heuristic methods. Essentially in this imperfect world, to be eligible for mating, you need to be a realistic candidate—you need to be good, but cannot possibly be perfect. You cannot pretend or design to be too good- that may either lead to outright rejection (strict mapping), or your initiative might bear fruit, with you being allowed to mate and mutate and be given the chance to bring up a supposedly good offspring following the rules of the game.

We know now the realistic lower limit. We know vide Theorem 1 that we can only be as good—we cannot be any better than this lower bound. That, in the max norm, the closest we can get to the desired periodicity is given by this lower bound. The Genetic Algorithm will endeavor to attain this lower bound, to achieve the best possible smoothness. However, since we are essentially operating in a blindfold manner, having a quality measure, too good to be true (lesser than the lower bound of Theorem 1), is indeed a cause of disbelief, but not rejection of the candidate individuum. Within an admissible range to be experimentally determined, it is likely, that individuuums in this range, on both sides of this lower bound, will prove equally fit to converge to this lower bound, over successive populations, undergoing crossover and mutation.

Now let us see if our GA can be endowed with some guidance in the other direction. Namely, what guarantee can we seek to motivate ourselves to keep seeking a solution approaching the desired lower bound set in Theorem 1. Theorem 2 below comes in very handy here. It gives the essential guarantee of the existence of a solution using the authors' tree based method to produce a perfectly periodic schedule with quality factor no worse than the given upper bound. Could we ask for more? As long as the existence of a periodic schedule instance within the quoted upper bound stands, our continued search within our maturing GA population stands vindicated. Thus even though, we know, we are groping in the dark, we are sure that the object of search exists. Our friendly tree based algorithm, has taught us not to despair, even when far removed from the expected solution, for it knows, it can construct a solution within a better-bounded interval. That's endowing some sight to the essentially blind Genetic Algorithm.

Theorem 3, proved in [1] as quoted below substantiates our above desire and claim:

Theorem 3: For any instance J of the periodic scheduling problem, there exists a schedule S such that

$$C_{\text{MAX}}(J,S) = 1 + 3(2R_J)^{1/3} + O(R^{2/3}) < 1 + 3.78R_J^{1/3} + O(R^{2/3})$$

7.2 Adaptation of Theorem 2 and 3 in GA framework

The intelligent fitness function incorporates the hindsight contained in the quoted two theorems over and above the criteria mentioned in section 4. In particular, a score factor inversely proportional to distance from the lower bound mentioned in Theorem 1, is added to the score as calculated in the naïve implementation. Linear scaling of this distance-based score is maintained between l and L and is given a quadratic decay after L. This is to encourage successive populations to mate and cluster towards the lower bound L.

The fitness cannot definitely be (better) lower than l . On the other hand, the tree-based algorithm is sure to produce a solution within bound L . The GA with its sufficiently diverse population, the mechanism of selective mating and crossover, and the hindsight as endowed to it courtesy the above two results, lets is more confidently churn itself to seek a good reservation.

Fig.4 summarizes the search space partitioned, with respect to the lower and upper bounds of Theorem 2 and Theorem 3. The fitness function now incorporates an adequately scaled value from those in the range indicated in the diagram below in the overall fitness as indicated earlier in the naïve implementation. The scaling of values will tend to give a higher weight to those in a close neighborhood of lower bound l , with sharp decline as we approach and exceed the upper bound L . The following figure better illustrates our adaptations of Theorem 2 and Theorem 3. l and L have respectively denoted the lower and upper bounds. The zone of most potent suitors would be concentrated around the lower bound. Zone 2 is the zone of promise as per the tree-based algorithms [1]. We need not really seek solutions in zone 3, though they still could be eligible candidates; when we know that better candidates do indeed exist in zone 2.

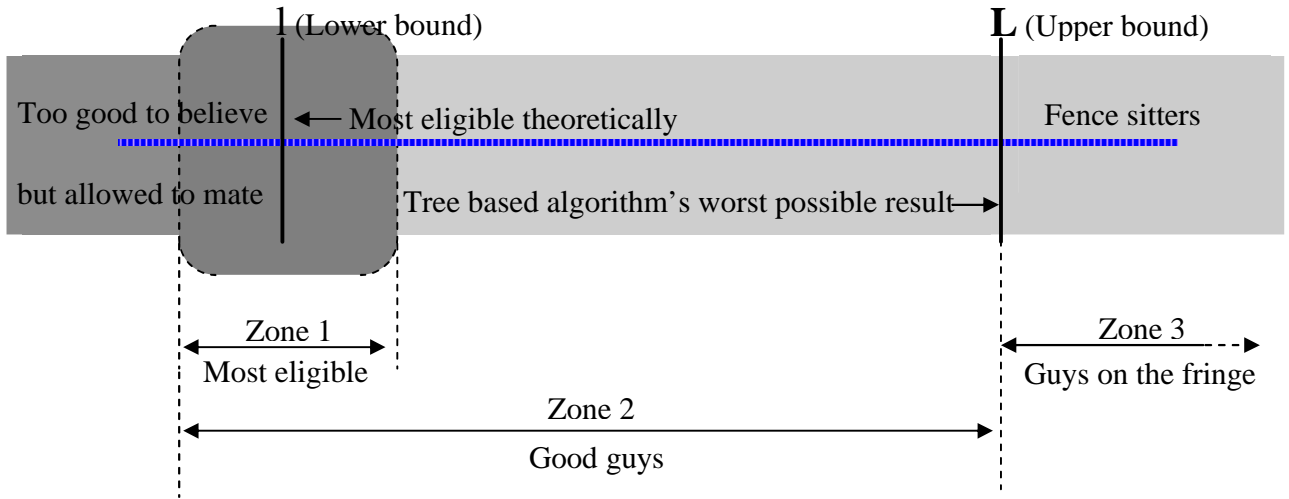


Fig 4: Individuum's eligibility zones

7.3 Adaptation of this modified slot scheduler for an AIS Network

As per section 1.5.1 of the IALA document [7], the shore station is responsible for filtering out slot reservation conflicts that may exist. These conflicts in the shore station network must be resolved separately from entering the data. The base station is not responsible for detecting these conflicts.

A base station in an AIS network will thus remain oblivious of the FATDMA slots of any neighboring base station. The design of the network software should thus ensure exchange of this slot scheduling information, and centralized contention resolution using either our method or some other. The key observation to be made here is that, this slot conflict resolution has to be a centralized operation, where data for each base station is available for contention resolution. This central server would thus do pre-processing like determining proximity between base stations, cluster formation etc. It is clear that stations far removed from each other can definitely have the same FATDMA schedule without jeopardizing the network integrity.

8. Conclusions and scope for future work

The algorithm needs to be tested over a large sample space of real data and the statistics of quality factor of resultant granted instances against input request instances to be generated. To study empirically the effect of different mutation operators and different crossover methods like 2-point cross over and other variants, as also the effect of number of individuals in the initial population etc. The results obtained in our preliminary testing seem to suggest that the obtained instances of periodic schedules for a query request come close to the lower bound proved in [1]. The correct zone size for zone 1, needs to be empirically determined with more experimentation.

Acknowledgements

I would like to thank the forward-looking management of NorControlIT, for nurturing an environment conducive for active applied research and product development. I would like to thank my wife Yogini Sahajpal, my brother Anurag Sahajpal, and my family for their support in my career and research endeavors. I hereby gratefully acknowledge the meticulous editing help rendered by Mr. Ashwin Chore for the various diagrams and also in formatting the mathematical expressions.

References

- [1] ZVIKA BRAKERSKI; AVIV NISGAV; BOAZ PATT-SHAMIR (2002), *General Perfectly Periodic Scheduling*, PODC 2002, Monterey, pp.163-172.
- [2] GOLDBERG, D. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley.
- [3] NOSSAL, R.; GALLA, T.M. (1997), *Solving NP-Complete Problems in Real-Time System Design by Multichromosome Genetic Algorithms*, Vienna University of Technology
- [4] BAR-NOY, A.; BHATIA, R.; NAOR, J.; SCHIEBER, B. (1998), *Minimizing service and operation cost of periodic scheduling*, 9th Annual ACM-SIAM Symp. on Discrete Algorithms, pp.11-20.
- [5] SAETHER, E.T. (2003), *High integrity computer system in student satellite NCUBE*, Diplomoppgave, Institutt for teknisk kybernetikk, NTNU, NORWAY.
- [6] "IALA / AISM IALA TECHNICAL CLARIFICATIONS ON RECOMMENDATION ITU-R M.1371-1 Edition 1.3"
<http://www.iala-aism.org/web/pages/publications/docpdf/ais/ITU.pdf>
- [7] "RECOMMENDATION ON AUTOMATIC IDENTIFICATION SYSTEM (AIS) SHORE STATION AND NETWORKING ASPECTS RELATING TO THE AIS SERVICE"(2002)
<http://www.iala-aism.org>
- [8] ACHARYA, S.; ALONSO, R.; FRANKLIN, M.J.; ZDONIK, B. (1995), *Broadcast disks: Data management for asymmetric communications environments*, Proc. 1995, ACM SIGMOD, pp.199-210.

A Practical Approach for Ship Construction Cost Estimating

Jonathan M. Ross, Proteus Engineering, Anteon Corporation, U.S.A., jross@anteon.com

Abstract

To succeed commercially, shipyards must be able to accurately estimate costs. Cost estimating is necessary for the bid process, for change orders, and for trade-off studies. Numerous cost estimating approaches exist. They are based on extrapolations from previously-built ships, detailed bottoms-up parametrics, and integrated physics-based analyses. Cost estimating can be frustrating to shipyard personnel. Cost estimators may lack timely technical information and face data inconsistencies. Ship engineers and naval architects commonly lack feed-back on the cost consequences of their technical decisions. Managers often lack information denoting the level of confidence in cost estimates upon which they must make business decisions. Finally, many approaches to cost estimating are mysterious and not formally validated (each cost estimator has his own black book), complicated (too time consuming to be of use to decision makers), or difficult to use (steep learning curve). This paper presents an approach that is simple, yet enables instant sharing of cost and technical data among ship engineers, naval architects and cost estimators; gives confidence measures to managers; and is user friendly. The approach is based on several years' development work in support of shipyard engineers, naval architects, and cost estimators.

1. Introduction

Most readers of this paper tend to view ship construction from a technical perspective: requirements, design, engineering, analysis, production planning, and production. Sometimes forgotten is the fact that ship construction is a businesses venture and must succeed financially as well as technically. This paper focuses on a key financial aspect ship construction: estimating costs.

1.1. Background

The ability to estimate ship construction costs is necessary for the commercial success of a shipyard; too high an estimate will place the shipyard out of the competitive range and too low an estimate will result in a financial loss and possible bankruptcy. In practice, an approximate cost estimate is developed during initial discussions with a potential customer. This estimate is refined as the discussions progress and the customer's requirements are defined in greater detail. The refined requirements result in higher levels of technical detail (e.g., concept design, preliminary design, contract design, and a specification of increased detail), which enable increased accuracy of the cost estimate. This process culminates in a cost estimate upon which the shipyard can base a fixed price bid.

Developing and refining a cost estimate is a complex and time-consuming process. Obstacles to success include faulty technical information (e.g., obsolete, incomplete, inconsistent), lack of communication among departments (e.g., rivalries, lack of peer-to-peer communication channels, secrecy), lack of a clearly defined process (e.g., ill-defined lines of authority, no freeze dates on design versions, different data formats), and problems with analytical tools (e.g., incompatible software, varying levels of detail, lack of features, too complex, not user friendly, not capable of being tailored to the needs of the shipyard or to specific projects). Overcoming these obstacles and producing viable cost estimates requires knowledge and skills of management, vendors and, most importantly, numerous shipyard departments, including engineering, production, planning, estimating, and marketing.

1.2. Scope of ship cost estimating

Ship cost estimating in general is a wide field, with the scope and depth of a given estimate tailored to

meet the needs of the user. Typical examples include the following:

- Construction (acquisition) costs – shipyard labor and material costs for design/engineering, production and testing
- Life cycle costs – construction costs plus maintenance, operation, support, and modernization
- Total ownership costs (applicable to naval ships or certain large commercial fleets) – construction, life cycle costs plus infrastructure costs for training, and other indirect costs.

Although this paper addresses cost estimating for new construction, a similar approach may be used to estimate costs for major repairs, overhauls, modernization, and disposal.

1.3. Example Approaches to Cost Estimating

Approaches to cost estimating vary from the informal to the formal, as described below:

- “Black book” – cost estimators create formulas, tables, and charts based on years of experience, industry trends, and vendor data. Typically, estimators guard this information closely, thus making its accuracy difficult to confirm. The black book approach can produce acceptable results in cases where the shipyard constructs a single or a few ship types and sizes. This approach is not so dependable for ship types or sizes beyond those normally constructed at the yard, or as costs become outdated.
- Parametric approach – System and subsystem costs are characterized in a spreadsheet or cost estimation program as a proportion of overall metrics such as length, volume, displacement and propulsion power. The proportions are estimated through comparisons with similar ships. As with the black book approach, if correlation levels are high, then the parametric approach yields good predictions; otherwise, the estimates may not be sufficiently accurate for many technical and business decisions.
- Standard ship approach – Some shipyards offer standard ship designs for which cost characteristics are well known. This enables the yards to very quickly and confidently develop detailed bids for prospective customers, and is an excellent solution if the designs match the customers’ requirements. However, even with the flexibility for making limited changes to the design, many customers prefer to purchase a ship that is more closely aligned to their business needs.
- Direct analysis approach – As the design matures, costs may be estimated based on drawings, bills of materials, historical vendor costs, and existing quotes. This approach is only practical after the design has reached a level of significant technical maturity.

Shipyards may use combinations of the above approaches. For example, the parametric approach may be used for structure, but the engineering approach may be used for owner-specified engine and auxiliary equipment. Cost estimates may be carried out by hand, spreadsheet, or on a computer program, and analysis results may be presented at various levels of detail.

1.4 Organization of this paper

The remainder of the paper presents a practical approach for ship construction cost estimating and is organized as follows:

- Cost estimating approach requirements
- Cost estimating approach description
- Case study using the cost estimating approach
- Conclusions
- References.

2. Cost Estimating Approach Requirements

“The cost estimating approach” presented in this paper complies with the following requirements:

- Three-tiered hierarchy of cost estimates to reflect varying levels of detail available to the cost estimator during the design process

- Each tier is independent of the others, permitting the best information to be used at all times, and not requiring that the estimate adhere to the “lowest common denominator” of information
- Material and labor elements are included (some shipyards may desire labor hours instead of labor cost because of confidentiality concerns)
- Confidence levels are presented to reflect the perceived accuracy of the engineering data and the cost estimating relationships.

Each of these requirements is discussed below (*Ross 2002*).

2.1 Three-tiered hierarchy

The cost estimating approach is divided into three tiers in order to reflect the three design phases (concept, preliminary, contract) commonly encountered in ship construction, Fig.1. At times, data may be available at different levels, and thus be placed in different tiers (see the following section, “Independence among tiers”). However, the norm is for data to be of a fairly consistent level of detail among the various parts of the ship (e.g., structure, propulsion, electric plant) during a given design phase. Thus, the corresponding tier is populated with technical and cost data during each of the three design phases, as described in the following paragraphs.

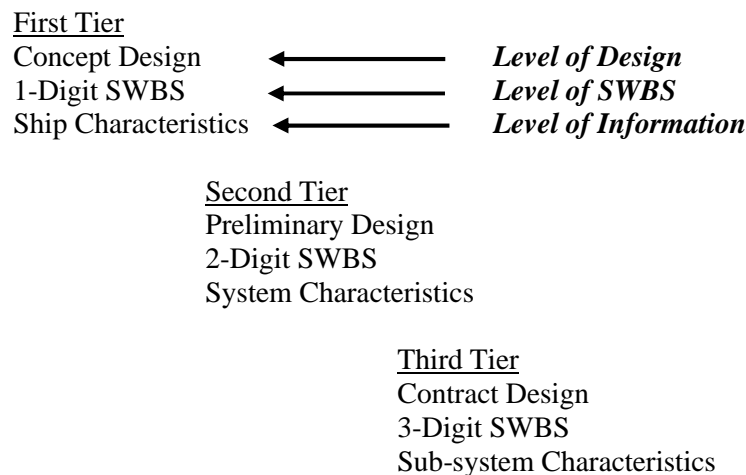


Fig.1: Three-tiered hierarchy of ship construction cost estimating

The first tier is Concept Design and is the least detailed. Typically this tier is used at the start of the cost estimating process when only limited information is available to the shipyard. This corresponds to elements of a 1-digit ship work breakdown structure (SWBS) and is based on about 20 data elements relevant to the whole ship (e.g., length overall, displacement, propulsion power).

The second tier is Preliminary Design. This corresponds to elements of a 2-digit SWBS and is based on about 125 data elements relevant to ship systems (e.g., structural system, propulsion system, heating and ventilation system).

The third tier is Contract Design and is the most detailed. The cost bid is based on the information in this tier. This tier corresponds to elements of a 3-digit SWBS and is based on hundreds or thousands of data elements relevant to sub-systems (e.g., main engine cooling, main engine fuel pumping, main engine starting).

2.2 Independence among tiers

Independence among the tiers allows the cost estimator to develop the estimate based on technical data of varying degrees of detail. For example, hull structure data may be available at the 3-digit SWBS level (e.g., 117 – Transverse Framing), but propulsion plant data may be available only at the

2-digit level (e.g., 220 – Engineering Control Systems). The cost estimating approach will accept data and produce reports for each of these SWBS levels. Thus, it is not necessary to wait until 3-digit propulsion data is available before populating structures at the 3-digit level of detail. Independence among tiers enables the cost estimate to be based on the most detailed (and presumably the most accurate) data available.

2.3 Material and labor included

Shipyards commonly divide costs into material and labor (material includes vendor and subcontractor costs, and labor is only that of shipyard employees). In order to best serve the shipyard needs, the cost estimating approach follows this convention by producing estimates for material and labor. Material estimates are provided as costs, but labor estimates are provided as labor hours (to maintain confidentiality of shipyard labor rates). Estimates are provided for each SWBS element for which technical data is available.

2.4 Confidence levels

Shipyard management needs to know the level of accuracy of the cost estimate in order to properly develop the bid. Put another way, management needs to know the level of uncertainty of the estimate. Uncertainty may be quantified either through the application of margin or the provision of confidence levels. Both are commonly subjective, though probabilistic calculations may be used.

The cost estimating approach uses confidence levels instead of margins. This is because confidence levels provide the user (e.g., management) with quantified insight into the accuracy of the estimate. With this knowledge in hand, if certain parts of the estimate have low confidence levels, then attention may be focused there to increase confidence levels, and thus increase the accuracy of the estimate. Confidence levels are assigned to the engineering quantities (e.g., reflecting a 90% confidence that the weight of structure is correct as reported) and also to the cost estimating relationships (e.g., reflecting 95% confidence in the estimated cost per weight factor). The two confidence levels are multiplied to arrive at an overall confidence level (e.g., $90\% \times 95\% = 86\%$). Confidence levels are presented by SWBS element.

3. Cost Estimating Approach Description

Shipyards commonly develop ship designs in the engineering department and develop ship cost estimates in the cost estimating department. The cost estimating approach is designed to support this division of labor, and was developed through input from engineers, cost engineers, and cost estimators. Key to the development success to date were workshops at which shipyard technical and cost personnel suggested enhancements to early versions of the cost estimating approach.

3.1. Description of the architecture

The cost estimating software is divided into two linked elements, one focused on engineering and the other focused on cost. Each element is comprised of modules which carry out discrete operations. A functional flow chart of the software is shown in Fig.2.

The engineering element comprises the following five modules:

- Baseline ship engineering quantities – This module is used in the case where the design ship (i.e., the ship for which cost is being estimated) is an extrapolation of a baseline ship (i.e., a ship for which costs are known). This module is populated with data which describes both ships in terms of selected physical quantities (e.g., tonnes of structural steel).
- Baseline and design ship principal particulars – Again, this component is used if there is a baseline ship. The module is a repository for general (non-SWBS) data such as length between perpendiculars.
- Parametric engineering quantities – Parametric calculations are carried out in this module to

estimate design ship engineering quantities, based on a constant times the ratio of corresponding design and baseline ship principal particulars (e.g., [constant] x [design ship length overall] / [baseline ship length overall]).

- Assigned engineering quantities – As naval architects and marine engineers develop the design, increasingly accurate engineering quantities become available for use in the cost estimating process. These “assigned quantities” are entered into this module. Normally, these engineering quantities are more accurate (higher confidence level) than the parametric engineering quantities of the previous module.

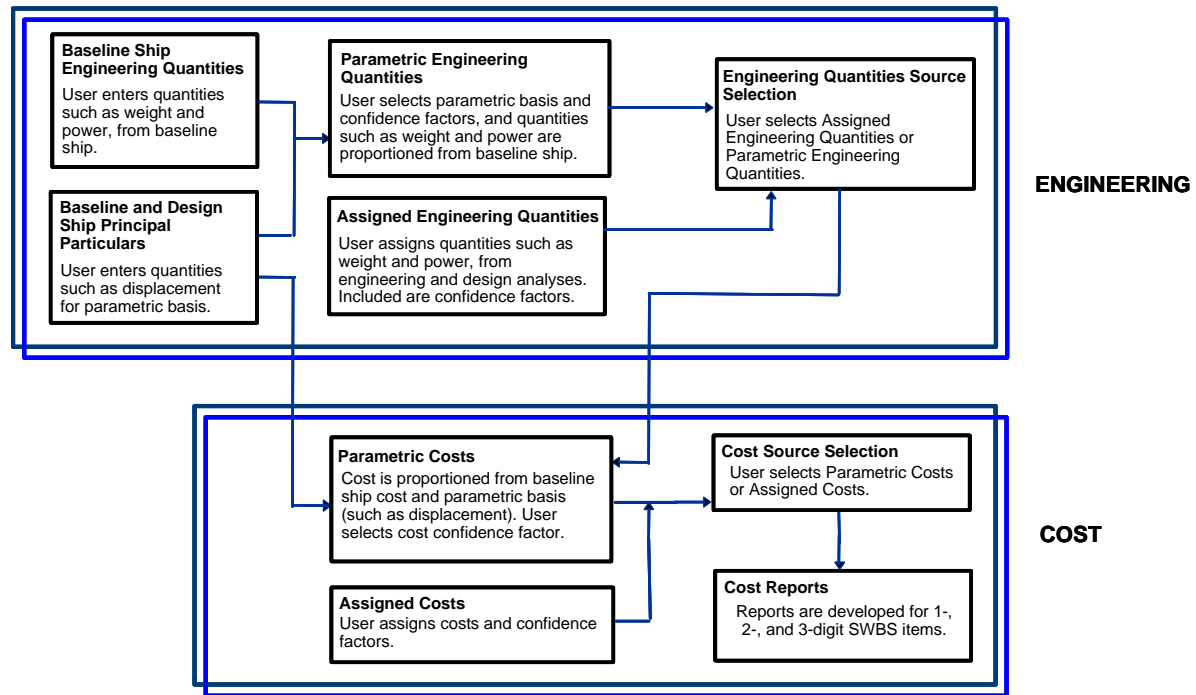


Fig.2: Flow Chart of Cost Estimating Approach

- Engineering quantities source selection – Here the user selects which engineering quantities (parametric or assigned) will be used in the cost estimating process. Normally, at the start of the design process, parametric engineering quantities are selected, and as the design progresses, assigned engineering quantities are selected.

The cost element comprises the following four modules:

- Parametric cost – As with parametric engineering quantities, cost is estimated for the design ship based on a proportionality with regard to the baseline ship.
- Assigned costs – As with assigned engineering quantities, assigned costs are directly entered into the module. These costs are based on data such as initial estimates from vendors and from purchase orders.
- Cost source selection – Again, as with the parametric engineering quantities source selection, the user selects between the parametric and the assigned values.
- Cost reports – This module produces three reports: 1-digit, 2-digit, and 3-digit SWBS cost estimates, with overall (engineering and cost) confidence levels provided for each cost entry.

3.2 User interface and data entry

The cost estimating software is hosted by a smart product model, to which various other components besides cost estimating may be added (e.g., structures, stability, hull form), *ROSS et al. (2001)*, *Ross, (2002)*. Engineering quantities, parametric constants, confidence levels, and cost data are entered and

reviewed in dialogue boxes and on Excel worksheets all of which may be tailored to the needs of specific users. Although the cost estimating software may operate in a stand-alone mode, the smart product model host offers several advantages:

- Integration of cost with engineering models – the engineering basis of the cost estimates (e.g., weight, volume, engine power) can be automatically linked to receive input from the engineering components.
- Enhanced communication within the design team – engineers, cost estimators, production planners, and management view a consistent set of technical and cost data.
- Configuration control – permission can be assigned to ensure that only authorized users enter and revise data in their respective task areas.
- One-stop high level modeling and estimating – the engineering and cost information developed by the smart product model is at a level at which sufficient detail is available to develop data on which meaningful technical and cost decisions can be based, yet the quantity of data is small enough to permit quick side studies, and to reflect the present state of an ongoing design.
- Single database for engineering and cost data – data can be stored in one place for all aspects of the design and cross referenced, presented in various hierarchies, and formatted for reports. A single database helps ensure data consistency.

4. Case Study Using the Cost Estimating Approach

A case study will illustrate the cost estimating approach with a double hull tanker as the example ship. The tanker is similar to the Double Eagle tanker design of Newport News Shipbuilding (now Northrop Grumman Newport News) shown in Fig.3, with the ship's principal characteristics presented in Table I, *HATFIELD, 1999*. This tanker design was featured in Maritech ASE Project 21, *DUNCLIFT, 2001*. Slightly revised versions of the Double Eagle SWBS and weight breakdown are used in the case study.

4.1 Cost and labor hour conventions

All costs and hour information and certain weight information are notional, and are not based on actual Double Eagle data. With regard to cost, much depends upon where (country and shipyard) the ship is constructed. Actual costs are not as relevant to this case study as showing the cost estimating approach. Thus, an approximate cost is considered sufficient. *BROWN, 1996* quotes \$39,400,000 for constructing a 40,000 dwt double hull tanker, based on *National Research Council, 1991*. Accounting for the increased deadweight of the example tanker and assuming a 3% annual inflation rate, the cost in 2003 would be \$64,200,000. *MARINE LOG, 2003* listed a pending contract for three 45,000 dwt crude/product tankers at \$68,000,000. The average of those two estimates (\$66,000,000) will be used for the case study (estimated construction cost based on engineering drawings and input from vendors).

Half of the construction cost is assumed to be absorbed by shipyard labor, and the assumed labor rate is \$30/hour, resulting in total labor hours of 1,100,000. Subdivision of cost and hours is based on weight.

4.2 Cost estimating process

Fig.4 shows the host smart product model user interface. The “Tasks” pull-down menu is shown, with “Cost” highlighted. After selecting Cost, the user enters the cost module and proceeds through a series of forms to enter project information, enter and calculate engineering and cost data, and print reports. The process is described below.

- Project information applies to the project as a whole, that is, to management, engineering, and cost estimating, Fig.5.

- Project information form – the user enters project name, point of contact (e.g., project manager), date, ship type, and contact information.
- Baseline ship general information form and (not shown) design ship general information form – the user enters ship characteristics (e.g., length overall) that may be used in parametric formulas for estimating engineering quantities (e.g., weight of the main deck structure), material costs (e.g., cost of main deck structure), and labor hours (e.g., hours to construct main deck structure).
- Ship element hierarchy entry – in a tabular format, the user enters the desired ship element hierarchy (in this case study, a SWBS hierarchy)
- Engineering data and calculations apply to the quantities and confidence levels for baseline ship and design ship SWBS elements.
 - The user enters engineering quantities for the baseline ship in the tabular format shown in Fig.6. In this instance, weight in tons is used. Weight is normally preferred for SWBS Group 100 (structure), because weight will provide an accurate estimate of material cost. In other SWBS groups, other engineering quantities may be more appropriate; for example, power may be more appropriate for SWBS Group 200 (propulsion).
 - The user assigns design ship quantities if available in a like manner as for the baseline quantities, as shown in Fig.7. Normally, as the ship design matures, these quantities become available. They may be directly input from other linked components of the host smart product model. Note that at the beginning of the design process, this level of detail will normally not be available, and an estimate is made as shown in Fig.8.
 - The user selects either the assigned quantities or the estimated quantities for further use in the cost estimating process (form not shown).
- Cost data and calculations apply to the engineering quantities selected in the last step above, and to the costs associated with the baseline ship SWBS elements.
 - The user enters baseline material costs and labor hours in for each SWBS element in a tabular format, as above.
 - The user assigns design ship costs for each SWBS element in a tabular format. Again, these costs may become available only at the later stages of the design, and during the beginning of the design process, cost estimating relationships (CERs) are used as shown in Fig.9.
 - The user selects either the assigned or the parametric material costs/labor hours.
- Print reports for three SWBS levels of detail. The reports show the combined confidences of “engineering quantities confidence” and “cost confidence.” 2- and 3-digit SWBS reports employ roll-ups from the detailed to general levels (e.g., 1-digit SWBS elements are sums of their respective 2-digit sub-elements), Figs. 10 to 12.

Table I: Principal characteristics of example double hull tanker

CHARACTERISTIC	MAGNITUDE
Length between perpendiculars	180.0 m
Maximum beam	32.2 m
Depth (at side)	19.2 m
Design draft	11.2 m
Deadweight	45,700 t
Engine power (bhp)	13,390 hp
Speed	16 kt
Accommodations	35



Fig.3: Example tanker (HMI BRENTON REEF)(Northrop Grumman Newport News)

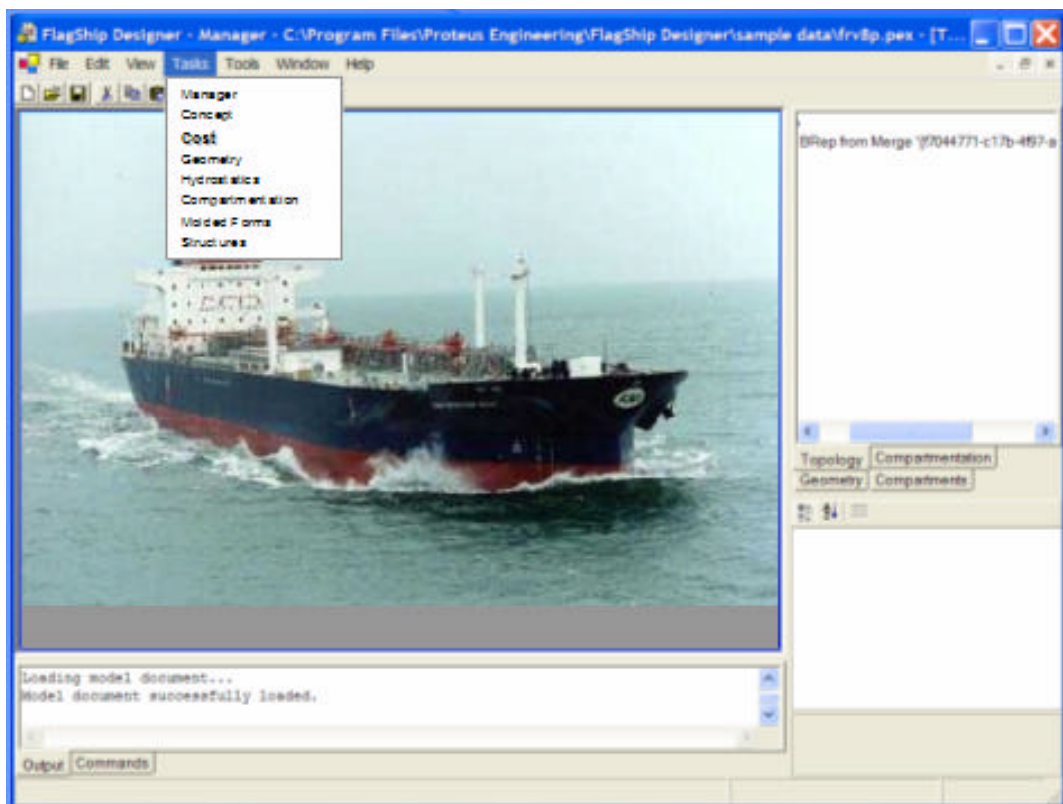


Fig.4: Selection of cost module in host smart product model

PROJECT INFORMATION FORM

Project Name
Double Hull Product Tanker

Point of Contact (POC)
Harry Smith

Ship Type
Tanker

Date
1/16/2004

POC Phone
...

PROJECT INFO

BASELINE SHIP GENERAL INFORMATION FORM

ITEM:
Length Between Perpendiculars

NAME:
BLP

UNITS:
M

METRIC:
LENGTH

QUANTITY:
180

Record: 14

SVBS NUMBERS AND CATEGORIES >>	SVBS NO	SVBS CATEGORY
	100	STRUCTURE
	110	SHELL AND SUPPORTING STRUCTURE
	111	PRIMARY STRUCTURE
	112	SHELL PLATING AND STIFFENERS
	150	DECK HOUSE STRUCTURE
	152	FWD DECKHOUSE SUPERSTRUCTURE
	153	AFT DECKHOUSE & FUNNEL
	160	SPECIAL STRUCTURES
	161	STRUCTURAL CASTINGS & FORGINGS
	167	HULL STRUCTURAL CLOSURES

Fig.5: Project information

SVBS NO.	SVBS CATEGORY	BASLINE MATERIAL QUANTITIES >>	BASLINE MTL QTY	BASLINE MTL UNITS	BASLINE MTL BASIS	QU
100	FANTAIL		10208.26	T	WEIGHT	
110	SHELL AND SUPPORTING STRUCTURE		9113.44	T	WEIGHT	
111	PRIMARY STRUCTURE		7754.16	T	WEIGHT	
112	SHELL PLATING AND STIFFENERS		1359.28	T	WEIGHT	
150	DECK HOUSE STRUCTURE		423.44	T	WEIGHT	
152	FWD DECKHOUSE SUPERSTRUCTURE		286.12	T	WEIGHT	
153	AFT DECKHOUSE & FUNNEL		137.32	T	WEIGHT	
160	SPECIAL STRUCTURES		179.89	T	WEIGHT	
161	STRUCTURAL CASTINGS & FORGINGS		43.76	T	WEIGHT	
167	HULL STRUCTURAL CLOSURES		25.09	T	WEIGHT	
168	SUPERSTRUCTURE STRUCTURAL CLOSURES		2.15	T	WEIGHT	
169	SPEC PURP CLOSURES & STRUCTURES		108.89	T	WEIGHT	

Fig.6: Baseline ship material quantity entry table

ASSIGNED DESIGN SHIP MATERIAL QUANTITIES >>	ASSIGNED DESIGN SHIP MTL QTY	ASSIGNED DESIGN SHIP MATERIAL CONFIDENCE	ASSIGNED DESIGN SHIP MTL UNITS	ASSIGNED DESIGN SHIP MTL BASIS	AS DES L QUAN
	11229.09	0.92	T	WEIGHT	
	10024.78	0.92	T	WEIGHT	
	8529.58	0.92	T	WEIGHT	
	1495.21	0.92	T	WEIGHT	
	465.78	0.92	T	WEIGHT	
	314.73	0.92	T	WEIGHT	
	151.05	0.92	T	WEIGHT	
	197.88	0.92	T	WEIGHT	
	48.14	0.92	T	WEIGHT	
	27.60	0.92	T	WEIGHT	
	2.37	0.92	T	WEIGHT	
	119.78	0.92	T	WEIGHT	
	10.73	0.92	T	WEIGHT	
	10.73	0.92	T	WEIGHT	

Fig.7: Assigned material tabular entry

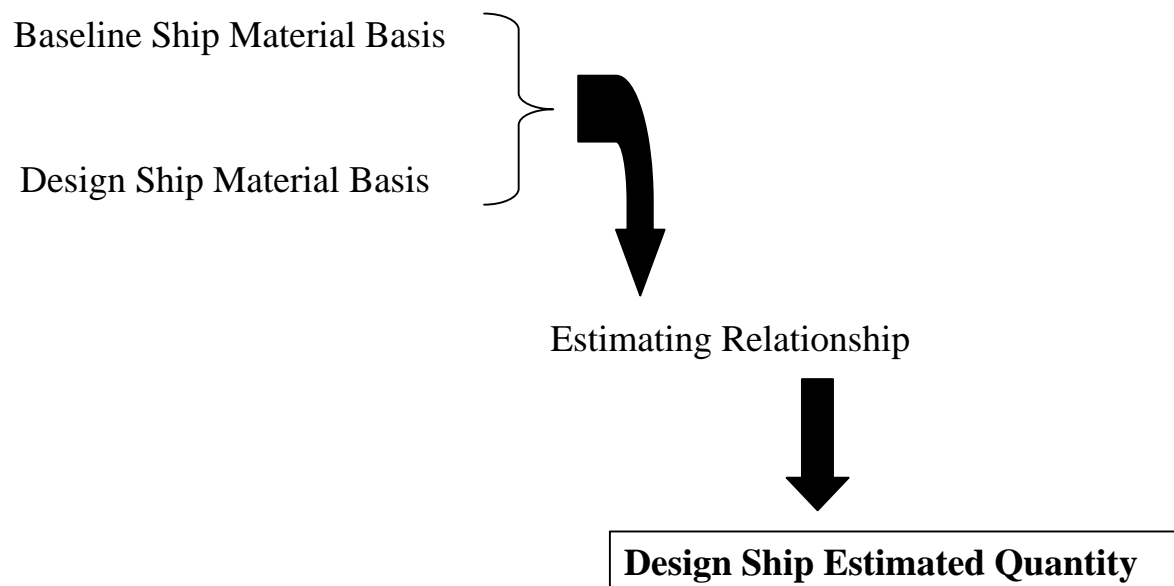


Fig.8: Approach to derive design ship estimated engineering material quantities

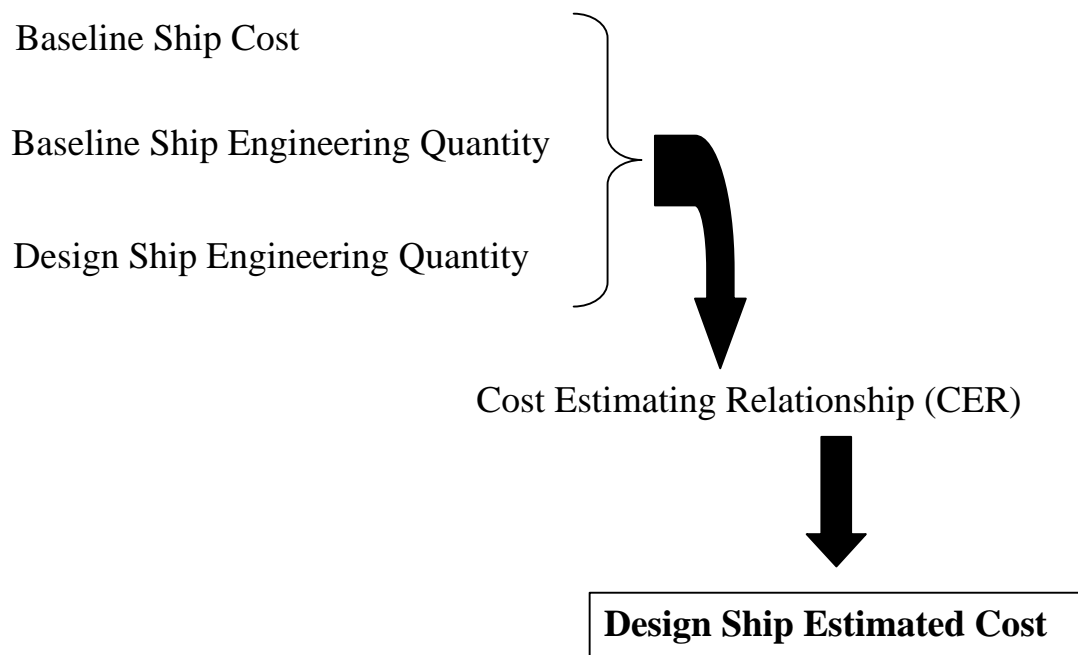


Fig.9: Approach to derive design ship estimated material costs and labor hours

DESIGN SHIP 1-DIGIT SWBS COST AND LABOR REPORT

PRICES

	MATERIAL DOLLARS			LABOR HOURS			
	COST	CON	TOL (+/-)	LABOR	CON	TOL (+/-)	
GRAND TOTALS >	\$ 29,871,725	0.96	1,194,869	995,724	0.96	39,829	
SWBS NO. AND CATEGORY							
100	STRUCTURE	\$ 24,614,665	0.96	\$ 984,587	820,489	0.96	32,820
200	PROPULSION	\$ 1,680,545	0.96	\$ 67,222	56,018	0.96	2,241
300	ELECTRIC PLANT	\$ 375,340	0.96	\$ 15,014	12,511	0.96	500
400	COMMAND AND CONTROL	\$ 41,208	0.96	\$ 1,648	1,374	0.96	55
500	AUXILIARY SYSTEMS	\$ 2,319,454	0.96	\$ 92,778	77,315	0.96	3,093
600	OUTFIT AND FURNISHINGS	\$ 840,513	0.96	\$ 33,621	28,017	0.96	1,121

Fig.10: Design ship 1-digit SWBS report

DESIGN SHIP 2-DIGIT SWBS COST AND LABOR REPORT

PEICES

MATERIAL DOLLARS			LABOR HOURS		
COST	CON	TOL (+/-)	LABOR	CON	TOL (+/-)
GRAND TOTALS >					
\$29,871,725	0.96	1,194,869	995,724	0.96	39,829

NOTE: ONE-DIGIT ENTRIES BELOW ARE ROLL-UPS OF TWO-DIGIT ENTRIES

SWBS NO. AND CATEGORY	MATERIAL DOLLARS			LABOR HOURS		
100 STRUCTURE	\$ 24,614,665	0.96	\$ 984,587	820,489	0.96	32,820
110 SHELL AND SUPPORTING STRUCTURE	\$ 21,974,781	0.96	\$ 878,991	732,493	0.96	29,300
150 DECK HOUSE STRUCTURE	\$ 1,021,020	0.96	\$ 40,841	34,034	0.96	1,361
160 SPECIAL STRUCTURES	\$ 433,760	0.96	\$ 17,350	14,459	0.96	578
170 MASTS, KINGPOSTS, AND SERVICE PLATFORMS	\$ 23,510	0.96	\$ 940	784	0.96	31
180 FOUNDATIONS	\$ 191,381	0.96	\$ 7,655	6,379	0.96	255
190 SPECIAL PURPOSE SYSTEMS	\$ 970,215	0.96	\$ 38,809	32,340	0.96	1,294
200 PROPULSION	\$ 1,680,545	0.96	\$ 67,222	56,018	0.96	2,241
230 PROPULSION UNITS	\$ 966,260	0.96	\$ 38,650	32,209	0.96	1,288
240 TRANSMISSION AND PROPULSOR SYSTEMS	\$ 121,358	0.96	\$ 4,854	4,045	0.96	162
250 PROPULSION SUPPORT SYS. (EXCEPT FUEL AND LUB)	\$ 285,733	0.96	\$ 11,429	9,524	0.96	381

Page 1

Fig.11: Design ship 2-digit SWBS Report

DESIGN SHIP 3-DIGIT SWBS COST AND LABOR REPORT

PEICES

MATERIAL DOLLARS			LABOR HOURS		
COST	CON	TOL (+/-)	COST	CON	TOL (+/-)
GRAND TOTALS >					
\$ 29,871,724	0.96	1,194,869	995,724	0.96	39,829

NOTE: ONE-DIGIT ENTRIES BELOW ARE ROLL-UPS OF TWO-DIGITS; TWO-DIGIT ENTRIES ARE ROLL-UPS OF THREE-DIGITS

SWBS NO. AND CATEGORY	MATERIAL DOLLARS			LABOR HOURS		
100 STRUCTURE	\$ 24,614,665	0.96	\$ 984,587	820,489	0.96	32,820
110 SHELL AND SUPPORTING STRUCTURE	\$ 21,974,781	0.96	\$ 878,991	732,493	0.96	29,300
111 PRIMARY STRUCTURE	\$ 18,697,217	0.96	\$ 747,889	623,241	0.96	24,930
112 SHELL PLATING AND STIFFENERS	\$ 3,277,564	0.96	\$ 131,103	109,252	0.96	4,370
150 DECK HOUSE STRUCTURE	\$ 1,021,020	0.96	\$ 40,841	34,034	0.96	1,361
152 FWD DECKHOUSE SUPERSTRUCTURE	\$ 689,907	0.96	\$ 27,596	22,997	0.96	920
153 AFT DECKHOUSE & FUNNEL	\$ 331,113	0.96	\$ 13,245	11,037	0.96	441
160 SPECIAL STRUCTURES	\$ 433,760	0.96	\$ 17,350	14,459	0.96	578
161 STRUCTURAL CASTINGS & FORGINGS	\$ 105,516	0.96	\$ 4,221	3,517	0.96	141
167 HULL STRUCTURAL CLOSURES	\$ 60,498	0.96	\$ 2,420	2,017	0.96	81
168 SUPERSTRUCTURE STRUCTURAL CLOSURES	\$ 5,184	0.96	\$ 207	173	0.96	7

Page 1

Fig.12: Design ship 3-digit SWBS report

5. Conclusions

The cost estimating approach has the potential to improve ship construction cost estimate accuracy and timeliness by:

- Increasing the ease by which technical and cost information is shared among shipyard users (e.g., management, engineering department, estimating department)
- Ensuring that appropriate technical information is provided to cost estimators in the correct quantity and format
- Providing a common cost estimating and reporting framework
- Functioning at a level of detail where changes may be quickly made.

References

BROWN, R.S.; SAVAGE, I, *The Economics of Double-Hulled Tankers*, 'Maritime Policy and Management', vol. 23(2), pp. 167-175, 1966

DUNECLIFT, L.A., *Develop and Implement 'World Class' U.S. Material Standards and Parametric Design Rules to Support Commercial and Naval Auxiliary Ship Construction*, 2001 Ship Production Symposium, The Society of Naval Architects and Marine Engineers, Ypsilanti, Michigan, United States

HATFIELD, M., *Newport News Shipbuilding's Final Double Eagle Tanker, HMI BRENTON REEF, Completes Successful Sea Trials*, <http://www.nn.northropgrumman.com/news/1999/nr990527.stm>

MARINE LOG, *Shipyard Contracts Awarded*, 'Marine Log', Simmons-Boardman Publishing Corporation, New York, p. 40, February 2003

NATIONAL RESEARCH COUNCIL, Committee on Tank Vessel Design, *Tanker Spills: Prevention by Design*, Washington, D.C., National Academy Press, 1991

ROSS, J.M. ; McNATT, T.R.; HAZEN, G. (2001), *The Project 21 Smart Product Model – A New Paradigm for Ship Design, Cost Estimation and Production Planning*, 2001 Ship Production Symposium, The Society of Naval Architects and Marine Engineers, Ypsilanti, Michigan, United States, Paper 6

ROSS, J.M. (2002), *Forging a Real-Time Link Between Initial Ship Design and Estimated Costs*, 11th International Conference on Computer Applications in Shipbuilding, Malmö, Sweden, pp. 75-88

Communication in Ship Design Networks

Robert Bronsart, Steffen Gau, Diane Luckau, Wolfgang Sucharowski

University of Rostock/Germany

{robert.bronsart,steffen.gau,diane.luckau,wolfgang.sucharowski}@uni-rostock.de

Abstract

The complexity of modern commercial ships in combination with short product development and production times forces shipyards to establish widely spread engineering networks. An analysis of distributed product development in these networks is presented. Networks are discussed in terms of participants, structure, timing and duration of collaboration processes. The investigations are based on a detailed analysis of actual newbuilding projects. Considered companies are shipyard, classification society, design agent and system/component supplier. Focus is given on both information exchange procedures and communicated information contents. Communication events in inter-organisational communication are identified and classified. Classification is done with respect to used communication techniques as well as quantity, frequency and timing. Typical problems encountered in communication processes are described. An other aspect presented addresses information contents and representations communicated between participants in collaborative ship design. CAE-systems in operation today are applications capable of handling complex product model data internally. Interfacing these applications by product data exchange has often been claimed. Shipbuilding projects analysed reveal the gap to ship design in today's practice. Collaboration based on product data exchange integrated into legacy applications is still rarely encountered. Low-level representations of existing product data are however exchanged frequently. Shared product data are still mostly based on drawings, exchanged using paper or files.

1. Introduction

Shipbuilding today is characterised by small series or one-of-a-kind production, fluctuating work load and fierce competition in terms of price, product delivery and quality. Design and production of modern commercial ships are complex processes. In order to realise ship newbuildings in time and quality, intense inner- and inter-organisational collaboration is required. Numerous specialised companies are co-operating in different stages of product design and production. Shared work has become a core element in today's shipbuilding industry in Europe. Its relevance is reflected by the fact that up to 70 % or even more of a shipyard's value creation is based on purchased equipment and services. This percentage obviously is a function of ship type, newbuilding project scheduling and the shipyard's business strategy. With increasing product complexity, like passenger vessels or offshore units, this figure can even be larger.

Design and fabrication time spans are becoming shorter; in addition the complexity of modern commercial vessels is increasing steadily. Taking these factors into account, shipyards are forced to establish engineering networks to run ship newbuilding projects in available time and to required quality. These engineering networks are formed by external companies, being involved in sub-phases of the overall shipbuilding process. Resources and expertise of these companies need to be assigned to individual engineering tasks, their output has to be integrated into the final product definition. The companies involved are of different type and are taking different roles. Some roles in the collaboration process are represented by dozens of companies while other roles are represented by one particular partner only.

2. Development and production process

The process of product development and production of ships comprises a number of different sub-phases, *ISSC (2000)*. As outlined above, the time span available to run development and fabrication

processes have become strictly limited. The result is an increased workload, exceeding the shipyard's own resources. Therefore an increasing amount of engineering tasks needs to be allocated to external resources. Assuming a constant total amount of resources required within a ship newbuilding project, reduced process time (Dur) causes an increase of input from external resources ($RAEx$ [%]):

$$RAEx = \frac{100(100 - Dur)}{Dur}; \quad 0 < Dur \leq 100$$

The formula is based on 100% shipyard resources resulting in project completion in defined 100% project period. It is assumed that the total amount of resources to run a ship newbuilding project is constant. As the shipyard's resources are kept to 100% the share of externally allocated resources needs to be raised accordingly if the available project time becomes shorter. As seen and analysed from today's shipbuilding practice, this approach underestimates the additionally required resources significantly: Any overhead for the co-ordination of externally allocated engineering tasks is not considered. Taking the required co-ordination efforts into account, the total amount of resources (TAR [%]) is considerably larger, it can be approximated to:

$$TAR = OV \cdot \frac{100(100 - Dur)}{Dur} + 100; \quad 0 < Dur \leq 100$$

where factor OV represents the amount of overhead for the co-operation management. According to the experiences made by the shipyards interviewed, this factor can reasonably be taken as 1.2 - 1.3 which means an additional 20 - 30% to the resources allocated to subcontractors. Apart from a constant value, an approach with a linear increase of the overhead based on the external workload can simply be included in the formula:

$$TAR = \left(1 + \frac{(100 - Dur)}{100}\right) \cdot \frac{100(100 - Dur)}{Dur} + 100; \quad 0 < Dur \leq 100$$

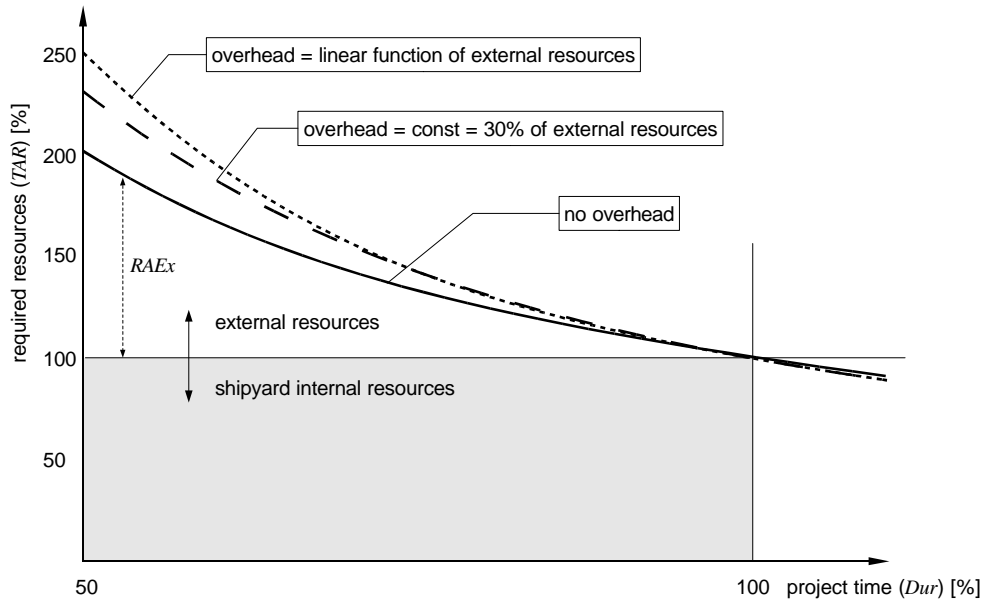


Fig.1: Required resources versus project duration

2.1. Design and procurement processes

Within the overall process of product design and production, different activities are to be distinguished. In the following, two classes are to be considered in more detail since these represent the majority of all inter-organisational collaboration: “*design*” and “*procurement*”. The relevance of these two classes is given due to the large numbers of corresponding communication events and the

number and volume of communicated information objects. An event in which communication takes place between two or more persons of different companies in the following is called “communication event”. An “information object” represents a document, holding any number of information in any format related to the product or the process, examples are report, drawing, data file.

Both classes of activities however show similarities. The fundamental difference justifying them to be treated separately is that procurement of ship systems and equipment activities in almost all cases finally result in delivery, processing, installation and integration of physical objects like steel plates, profiles, pipes, valves, diesel engines, pumps, thrusters etc. While managing a procurement class purchase process, a specific date exists, at which the collaboration results in physically realised objects (hardware). Layout, calculation, design and analysis activities however result in reports, drawings, spreadsheets, files or databases etc. (software).

2.1.1. Design

Activities of definition and design type are focussing on the definition of product features in terms of e.g. shape, materials, functionality, ... A shipyard functioning as general contractor to the customer is establishing an engineering network with other companies for the design of the product. In addition to this design centred co-operation with e.g. model basin and design agents other partners like classification societies are to be involved due to regulatory tasks like compliance checks and approvals. Design activities can be characterised by two main aspects.

1. Numerous different companies are collaborating representing different roles like owner, general contractor, sub-contractor, authority, supervisory body, see Fig 2. Partners participating have specific rights, responsibilities, obligations and dependencies. General terms and conditions of a classification society may serve an example, saying the classification society “... is an independent organisation..[-]... acts impartially and objectively ...[-]... Any information, drawings, etc. required for performance of the functions and activities ...[-]... *must be made available in due time.*” *GL (2004)*.
2. Due to simultaneous processes, product data are in many cases of preliminary state. The iteration like procedure is caused by the complexity of the product and the short time available in which numerous functional, fabrication and operational aspects are interacting. Therefore, many design tasks are to be started with intermediate input data. Revise or otherwise change of data results in multi-layered notification and re-work. Specific milestones e.g. design approval by the classification society exist on which numerous downstream design steps rely on.

2.1.2. Procurement

Compared to design, a much larger number of different companies is involved in procurement activities. Collaboration between shipyard and suppliers count for most activities of procurement type, *Bronsart and Gau (2001)*. In contrast to design, here objects in question get physical existence. Duality of paper-based and/or “electronic” object representation as well as physical object existence is challenging which causes special management requirements of procurement activities to guarantee coherence between all representations of the product. Specific communication scenario types have been identified, see below, with the ongoing project, communicating partners are changing frequently.

2.2. Process chain

The described classes of engineering activities are derived from generalised shipbuilding engineering tasks. It is obvious that both are interrelated with each other in many different ways. This can be seen from the fact that most types of collaborating companies are involved in both classes. Engineering tasks of either class are in permanent interaction in order to run the overall shipbuilding process successfully.

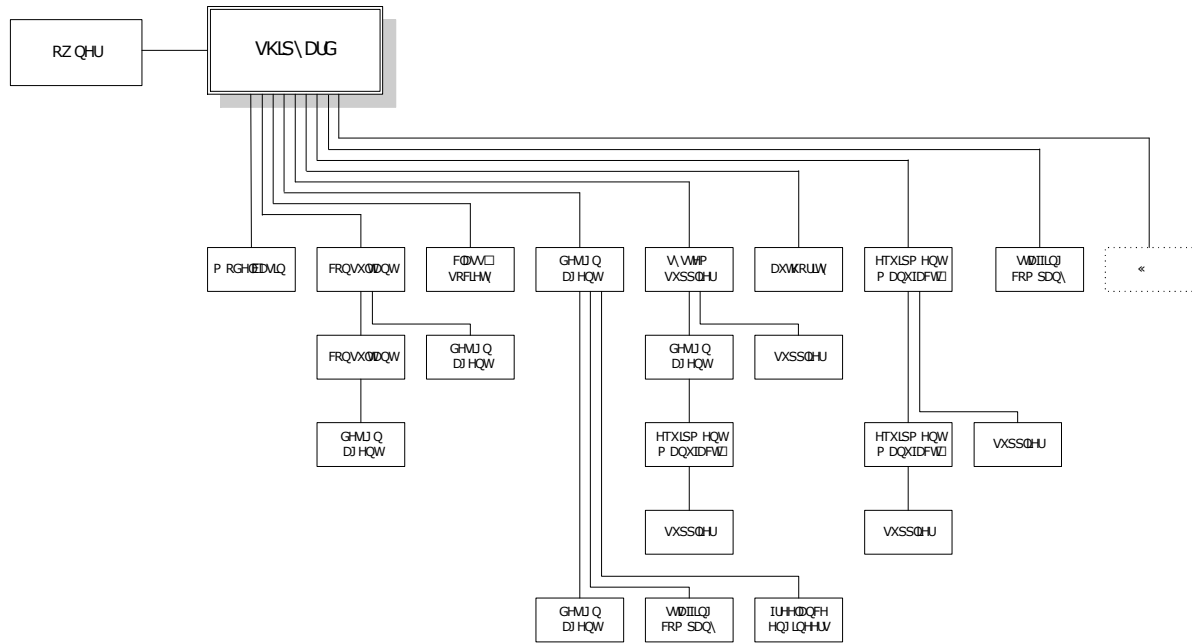


Fig.2: Multi-layered collaboration structure

Linkages exist in terms of multiple technical aspects, project scheduling, financial aspects and responsibilities. The initiations of numerous detail design tasks for instance succeed a formal steel structure approval by the classification society. Planning techniques are required to structure and control the multitude of tasks and their complex interactions. Different aspects, their combination and the types and number of external companies involved contribute to the complexity of the overall shipbuilding process. Typical companies subcontracted by a shipyard with further subcontractors are depicted in Fig.2 schematically.

3. Analysis of inter-organisational communication

The relevance of inter-organisational communication in shipbuilding has been explained. In order to understand the ongoing communication, a detailed analysis has been conducted. Data presented in the following are based on on-site analysis in different companies. Due to its importance for the overall process shipyard based investigations comprise the largest part of data. Collaboration projects of design class as well as procurement class have been analysed. As being the most important partners in terms of number and communication effort collaboration between shipyard and design agents as well as equipment supplier are taken into account, see Table I.

Table I: Overview of analysed collaboration scenarios

	partner		design	procurement	duration	ID
	on-site analysis	reference			[month]	
1	shipyard, design dep.	equipment supplier	+		20	A
2	equipment supplier	shipyard		+	14	B
3	shipyard, purchase dep.	equipment supplier		+	41	C
4	shipyard, purchase dep.	equipment supplier		+	32	D
5	shipyard, design dep.	equipment supplier	+		42	E1
	shipyard, purchase dep.	equipment supplier		+	26	E2
6	equipment supplier	shipyard		+	27	F
7	equipment supplier	shipyard		+	21	G
8	design agent	shipyard	+		14	H

The investigated collaborations are part of different ship newbuilding projects. For statistical reasons and to achieve rather industry specific than company specific data, newbuilding projects from different shipyards were considered. Ship newbuildings vary in size and type, ranging from container vessels to passenger vessels.

The duration period given in Table I is defined by the first and last communication event found in the documents at the partner where the analysis took place. Obviously due to the length of the duration in nearly all scenarios, the documented co-operation was relevant to at least a small series of ships. Scenario H however represents a co-operation in which a design agent was in charge of detailed design and production data generation for a single vessel, at the same time subcontracting part of the work to other design agents.

3.1. Communication media

Figures given in Table II and Fig.3 are based on a detailed scanning of the documentation on the inter-organisational communication at the partner's sites. For the categories face to face and phone, the indicated number of communication events reflects the partner's documentation strategy. For scenario H, the large number of communication events in a relatively small period of time (14 month) indicates a close and intensive co-operation between design agent and shipyard. The media distribution is depicted in Fig.3.

Table II: Communication media used in inter-organisational communication per scenario

media		A	B	C	D	E1	E2	F	G	H
letter	[-]	14	31	21	10	33	28	31	6	9
	[%]	21	26	10	9	21	29	23	3	1
e-mail	[-]	1	1	13	-	43	1	31	64	170
	[%]	1	1	6	-	28	1	23	35	21
face to face	[-]	3	2	6	4	4	1	6	-	20
	[%]	4	2	3	4	3	1	4	-	3
fax	[-]	47	74	158	88	73	61	53	104	404
	[%]	69	63	75	83	47	62	39	57	51
phone	[-]	3	10	13	4	2	7	14	8	191
	[%]	4	8	6	4	1	7	10	4	24
total no events	[-]	68	118	211	106	155	98	135	182	794

Remarkable is the large number of fax used during the communication process contrary to the number of e-mails. This can be explained by the still quite informal, nearly oral character of e-mail-communication.

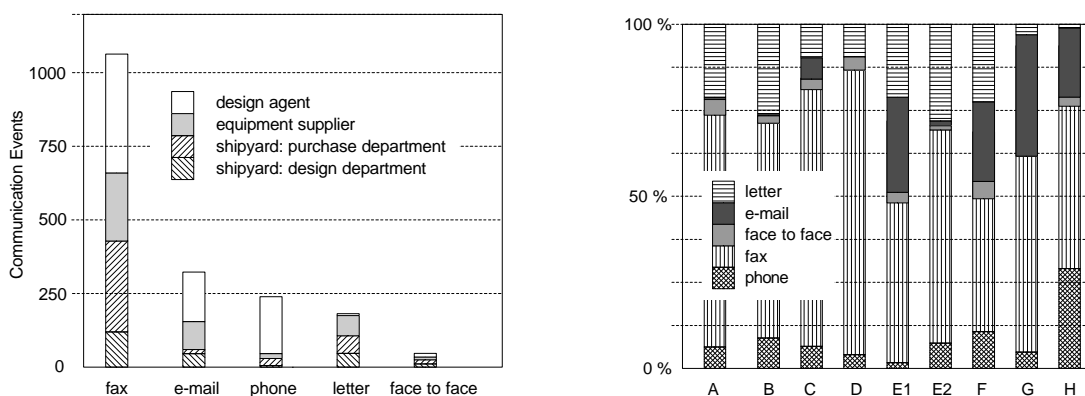


Fig.3: Media used in inter-organisational communication

E-mail is still regarded as relatively unreliable media. For confident contents, legally liable and binding information still letter or fax mail is preferred. Fax seems to be considered as more liable. For example, if there is no answer to a fax, another fax with the same content is sent. If there is no answer to an e-mail, the information usually will be resent by fax, not by e-mail again.

3.2. Media trends

According to Fig.4, media used changed during the last years. The number of oral communication events increases while the numbers of literal media decrease. Furthermore there is a general tendency to more informal as well as online communication methods. As once the fax has driven out the letter, now the e-mail is replacing the fax. No media is completely eliminated from the collaboration processes, but is used in a more specific way corresponding to a specific task. Every new communication method is only accepted by the users when it is considered as being more efficient than the means already in use.

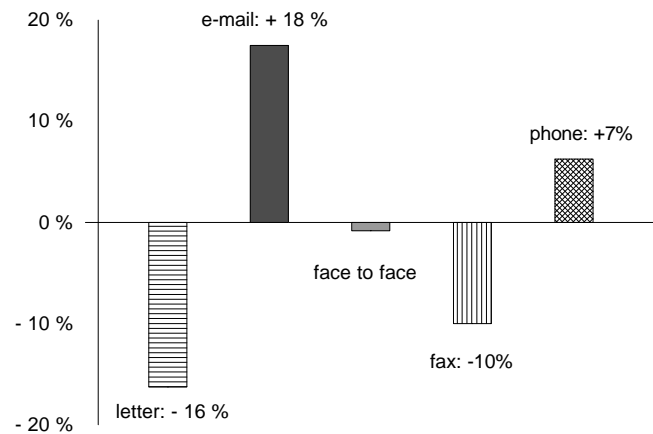


Fig.4: Trends in media use in the last decade

4. Communication patterns

Communication can be judged by the way it succeeds in fulfilling the intention of the sender/receiver in relation to the efficiency with which it is achieved. As can be seen from the analysis of the communication in shipbuilding networks, the priority task is to *manage* “something”. The second is to get it done with the least amount of energy and time. Therefore short reply times and a small number of communication events are indicators for a successful communication.

“Request - response” communication events consisting of a first step, a request and a second step, a response to the specific request are found in many cases. Furthermore communication events exist in which a direct answer is not expected or intended, single step. Considering that each of the two step events has a sender and a receiver, two, three or four different persons can be involved in this process. The simplest form is in which the receiver of the first communication sends the answer personally back to the sender, sender and receiver simply change their roles (type I). Often the receiver hands the information (and the responsibility) over to someone in the company who is regarded better versed in the specific subject and who sends the information back to the original sender, this is represented in type II. It also occurs that the receiver of the first communication doesn’t send the information back to the original sender, but to someone else in the senders company as shown in type III. Type IV however represents communication events in which the receiver of the first communication hands the information over to someone in its own company who sends the answer back to the other company but not to the original sender.

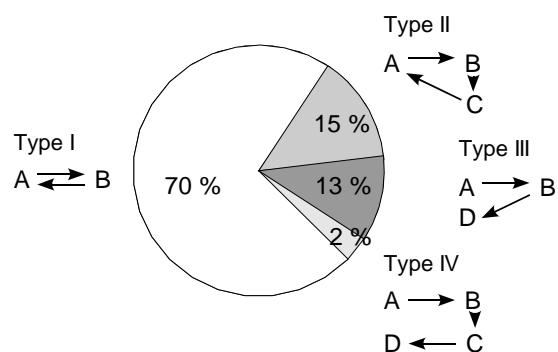


Fig.5: Communication patterns

The four principle communication patterns found and the distribution of these patterns with respect to all communication events identified are depicted in Fig.5. Considering that only replies of type I and II reach the original sender (A), approximately 15% of all replies are sent to someone else. A closer look on the companies involved in the communication process reveals some differences in the reply procedures. As can be seen from Fig.6, right part, in each company the proportion of type I communication is commonly at about 70%. In the process phases in which purchase of equipment is considered (supplier and shipyard purchase department), type II communication dominates type III. In design process phases, indicated by design agent and shipyard design department, it is just the opposite. On the left side of Fig.6 the portion of two step communication events is given with respect to the total number of communication events recorded.

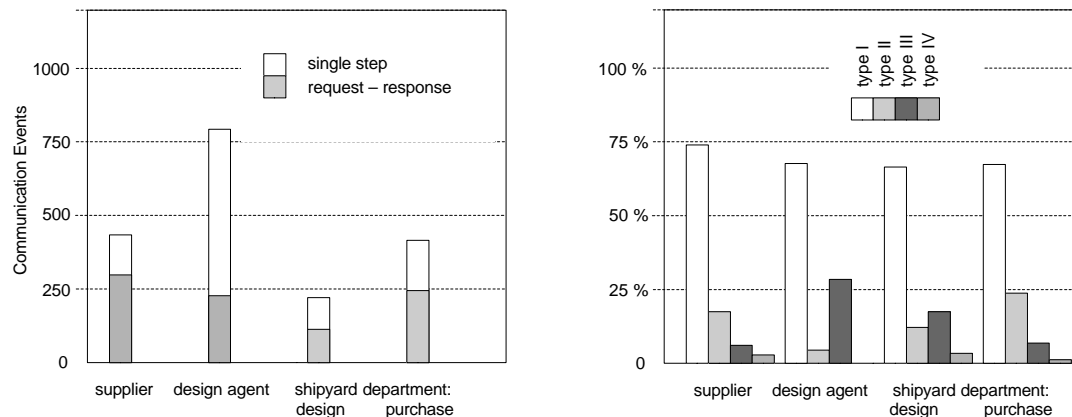


Fig.6: Communication patterns: partner/task details

Type II communication can scarcely be observed during the negotiation before completion of the contract. It is frequently used for technical details which cannot be answered by the purchase (sales) department; therefore the information is forwarded to the design department. Type III communication (the reply does not go back to A) happens either accidentally when the reply is sent to a “wrong” receiver or on purpose. The relatively large number of type III scenarios in the design process phase indicates that this type is used intentionally. By this way, the relevant information is distributed to several communication partners in the same company having to communicate intensely within the company. This in many cases complicates the process. In the sales or purchase departments this type is rather disturbing the usual process and has to be regarded to be accidental.

The turn around time in general for type I, II and III shows no differences as 50% of the requests are answered within the same day. It is necessary to have a closer look at the different companies taking part in the communication regarding the response times. In those processes where most of the communication events take place between the same two (sometimes three) persons in charge, the response times are clearly shorter than in those processes in which the functions A and B are represented by many different persons. On the average at least 35% of the requests are answered within the same day. One shipyard taking part in this study changed its communication strategy from various senders and receivers to one person functioning as a gateway in inter-company communication. Due to this reorganisation, the number of communication events in which the response was within the same day increased from about 30% up to 50%, in some cases to 80%. In case of such gateways are defined, type I communication is preferred.

It can be seen that constancy in communication partnerships has a positive effect on the response times. It is not important to know the specific person in the partner company, but to have a constant partner serving at the same time as gateway and distributor of the incoming communication. In this function the person at this crucial position needs detailed knowledge of the subject and the company structures to be able to send the right information to right places. At the same time this person is the one who surveys the whole process, it is the only one who has the complete overview.

5. Problems and bottlenecks

When starting the investigations one principal hypothesis was that misunderstandings and misinterpretations having their origins in verbal communication would affect the business communication and would cause delays or extra work. This hypothesis could not be confirmed by the data recorded. During the detailed investigations it became obvious that the language used in the communication process did not affect the process itself. No misunderstandings or delays could be found due to the use of different languages. The communicators are specialized on a specific vocabulary and its use. But if it is considered that once the contract has been signed a shipyard's only interest lies in having the ordered parts or services at the right time at the right place, the amount of communication while executing the externally allocated tasks was expected to be low.

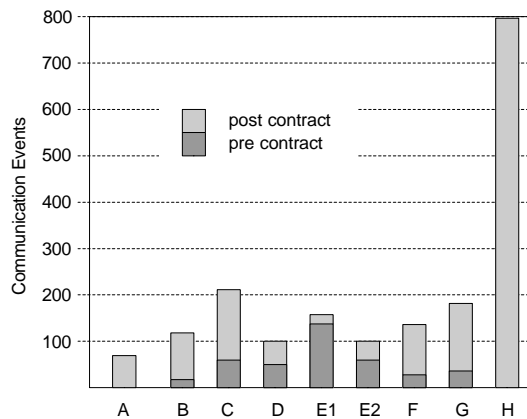


Fig.7: Communication events rel. to contract

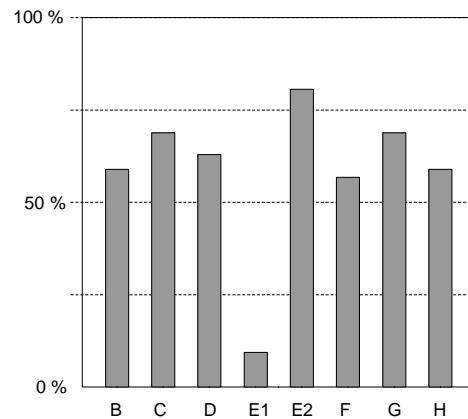


Fig.8: Communication due to disturbances

Communication volumes handled by a shipyard's purchase department during tendering and negotiation (pre contract) are almost constant. As projects C/D/E2 reveal these activities count for about 50 communication events, see Fig.7. Communication after signing the contract represents a major part in the process. The reason for this increased communication can be seen in Fig 8. With the exception of shipyard, design department (E1), 60 to 80% of the communication in the design process (post contract) consists of communication reporting and handling disturbances and reactions to it. Therefore the communication in this phase seems to be a trouble-shooting process. These disturbances may have their origins in the communication process itself such as no, incomplete or too late information transmitted, loss of information or parts thereof due to wrong media or formats used or in the selection of the wrong communication partner. The disturbances are also caused by changes in the product configuration and the iterative character of the overall design process.

For scenario H the distribution of communication events between design agent and shipyard over the collaboration period is shown in Fig.9 (~ 800 communication events in a 14 month period).

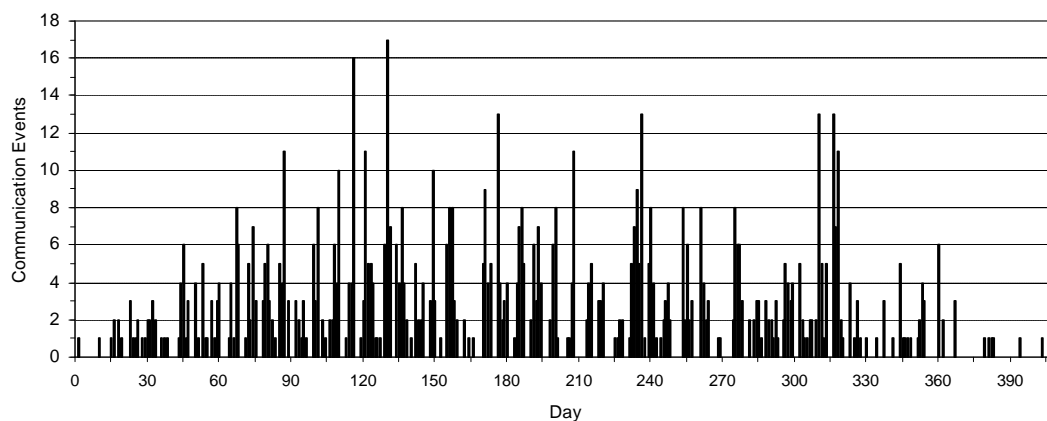


Fig.9: Profile of inter-organisational communication events for scenario H

6. Information content

In design class activities the product is viewed at in terms of technical data and information existing in different representations. Manufacture, supply and delivery details like packing, loading, shipment, off-loading, installation, commissioning and operation being recorded and found typical for procurement class collaboration processes are of no relevance. The total number of communication events has therefore been expected to be less than in the other class of activities. Furthermore the amount of product data exchanged “electronically” is expected to be far larger.

Table III gives a summary of communication events recorded and the types of exchanged technical information objects. Data shown in Table 3 are remarkable since volumes and distributions in practice are to some extent different to those expected. For design considerations projects A/E1/H are of interest. The total volume of communication events is not less than expected in comparison to procurement. The figures show that the assumption, the share of communication events exchanging technical data is larger is proven. An analysis shows that events in which technical data are exchanged count for 62/46/77% of all communication events. Drawings, either 2D product representations or spatial views are widely used and thus play a major role. 3D geometries and product data representations are still of less relevance.

Table III: Information content

communication events		A	B	C	D	E1	E2	F	G	H
total	[-]	68	118	211	106	155	98	135	182	794
technical data		42	12	19	12	71	9	31	38	608
of total	[%]	62	10	9	11	46	9	23	21	77
paper based	[-]	42	11	18	12	62	9	22	17	387
file based										
2D drawing	[-]	-	1	1	-	9	-	9	21	151
3D geometry	[-]	-	-	-	-	-	-	-	-	57
product model	[-]	-	-	-	-	-	-	-	-	13

The number of external companies involved in procurement of ship systems, components and equipment is significantly larger than in the design class of activities. Most external companies involved are component manufacturer and supplier. Their product portfolio is relevant to various industries, e.g. pumps, valves, fittings, electric drives. These suppliers perform in a purchase - supply relation to the shipyard. Exchanged information objects are inquiry, contract and handling documents, as well as technical data holding information on technical specification in form of dimensions, weights, connectivity etc. Large amounts of communication are caused by modifications of fundamentals, connectors, colour and labelling, delivery details and installation instructions.

Equipment suppliers for subsystems like e.g. power generation and distribution, HVAC, thrusters, shaft line, propeller, steering gear, mooring equipment, which are shipbuilding industry specific, are of greater relevance thus resulting in more complex dependencies and interactions with ship structures and on-board systems. Investigations show that even in these cases procedural information on e.g. scheduling or installation instructions count for most communication events taking place.

6.1. Detailed design scenario

In scenario H, a design agent is co-operating with a shipyard and other design agents to design hull structure details of a newbuilding. Participation in basic product definition asks for continuous dialog and co-operation with partners also involved in this process phase. Information, notification, request are typical situations initiating communication. Those activities are significant in early product definition, layout and basic design as the products complexity causes iterations. To notify others involved about current design states mainly paper based drawings and views are exchanged, media

fax is of significant importance. Paper based exchange of information objects namely drawings and views counts for 64% of all technical data exchange events. Exchange of files amounts to 221 communication events. Hull form geometry, surfaces and steel structure details communication events handling product model data are counting for about 12 % only. These events are associated with the delivery of detailed design results. The same applies for fabrication data linking design and manufacture.

Table IV: Details on exchanged information objects in scenario H

communication events	[-]	content type	shipyard		design agent	
			to	from	to	from
paper based drawing/view	387					
2D	341	arrangement	+	+		
3D views	46	spatial view	+	+	+	+
file based	221					
2D drawings/views	151					
hpgl	35	section plan	+			
dxf	95	view	+	+		
other	21	miscellaneous	+	+		
3D geometry	57					
dxf	50	arrangement	+			
iges	7	hull form	+			
product model	13					
hull structure details	4	database extract	+		+	+
hull structure	9	NC data	+			
production data						

Design agents as well as the shipyard are using shipbuilding specific CAE systems for almost all design tasks. These CAE systems, focussing on layout, definition, basic design, detail design, production planning and generation of production data, are based on complex internal data management. In order to export and import data complex procedures are required still causing remarkable efforts for both communicating partners. To enable external design work specific data are extracted, transferred and used for project initiation. After completion complex design data are transferred back. The majority of communication events are handled by transferring sub-sets of these data only.

7. Summary

Results presented are exemplary for collaboration scenarios in shipbuilding. Different company types playing different roles in the shipbuilding process are identified. Two typical classes of engineering activities are derived and discussed: design and procurement. Increasing outsourcing causes notable management and integration overhead, the overall process relies on well functioning communication methods. A large number of communication events are focusing on management functions e.g. tendering, negotiating, contracting, re-negotiating, requesting, notification. Since inter-organisational concurrent engineering relies on communication and co-ordination, the efforts to handle distributed network scenarios are decisive for the overall success. All externally supplied work needs to be initiated, managed, controlled and integrated. These efforts are remarkable and add to the primary contracted engineering tasks which count for numerous communication events.

Acknowledgement

The work presented in this paper is supported by the German Federal Ministry of Education and Research (bmb+f) under grant 03SX131. The authors acknowledge the valuable discussions with the partners of the “Shipbuilding Information and Communication Infrastructure” project and their support for the detailed analysis of communication procedures.

References

- ERIKSTAD, S.O., HAGEN, A. (2003), *Web-based tendering collaboration in project-centric industries*, 2nd International EuroConference on Computer and IT Applications in the Maritime Industries, COMPIT'2003, Hamburg, pp.255-264
- BRONSART, R.; GAU, S. (2001), *Collaborative design and production in shipbuilding networks*, 2nd International Conference Navy and Shipbuilding Nowadays, NSN'2001, St. Petersburg, pp.35-42
- BRONSART, R.; GAU, S. (2000), *A platform for co-operative inter-organizational work in design and production - Information management in collaborative shipbuilding*, 1st International EuroConference on Computer Applications and Information Technology in the Marine Industries, COMPIT'2000, Potsdam, pp.98-107
- BRONSART, R.; GAU, S. (1999), *An integrated design and production environment for ship machinery systems*, 10th International Conference on Computer Applications in Shipbuilding, ICCAS, Cambridge, pp.289-300
- GL (2004), *Rules & Guidelines 2004*, Germanischer Lloyd AG, Hamburg 2004, Section 1 E General Terms and Conditions, pp.1-2
- ISSC (2000), The 14th International Ship and Offshore Structures Congress, Report of Technical Committee IV.2 *Design Methods*, Nagasaki, 2000
- DÖRING, N. (1999), *Sozialpsychologie des Internet für Kommunikationsprozesse, Identitäten, soziale Beziehungen und Gruppen*, Hogrefe, pp.209-229. In German
- SCHWABE, G. (2001), „*Mediensynchronizität*“ – *Theorie und Anwendung bei Gruppenarbeit*, in: HESSE, F.W.; FRIEDRICH, H.F. (ed), *Partizipation und Interaktion im virtuellen Seminar*, Waxmann, pp.111-134. In German
- PICOT, A.; REICHWALD, R.; WIGAND, R.T. (1996), *Die grenzenlose Unternehmung. Information, Organisation und Management*, Gabler, pp.91-101. In German

A Multi-Agent based Approach for Solving the Vehicle Transhipment Problem

Torsten Fischer, Bundesministerium für Bildung und Forschung, Bonn/Germany, torsten.fischer@bmbf.bund.de

Hermann Gehring, FernUniversität Hagen, Hagen/Germany, hermann.gehring@FernUni-Hagen.de

Abstract

In this paper a multi-agent based approach for supporting the planning of transhipments of imported vehicles via a seaport automobile terminal is presented. The considered planning problem consists of two tasks, the allocation of parking areas for the temporary storage of vehicles and the allocation of drivers to the vehicles that have to be moved in the terminal area. These planning tasks are assigned to two different agent types of a multi-agent-system (MAS). A further agent, called coordinator agent, is composing the local subplans into an overall plan, so that the demand for drivers in the planning period is minimised and balanced. The MAS was subjected to a test using problem instances, which were generated in conformity with practical conditions. According to the test results, the MAS shows a robust behaviour with regard to changes in the problem data, in particular to the number of regular drivers and the cost surcharge for hired drivers. In addition, the dependency of the minimum of the overall (relative) costs of the drivers on the number of regular drivers and on the level of the cost surcharge for hired drivers is demonstrated.

1. Introduction and problem description

Finished vehicles imported into Europe are mainly transhipped via seaport automobile terminals. The incoming vehicles are stored on large paved parking areas until they are delivered to the carriers or shipping companies. Due to intensified transhipment processes, slot bottlenecks are occurring more frequently. In addition, the storage periods for imported vehicles are increasing. This imposes an additional load on the already restricted storage locations. If the regular storage locations are exhausted, excess vehicles are stored in a supplementary car park.

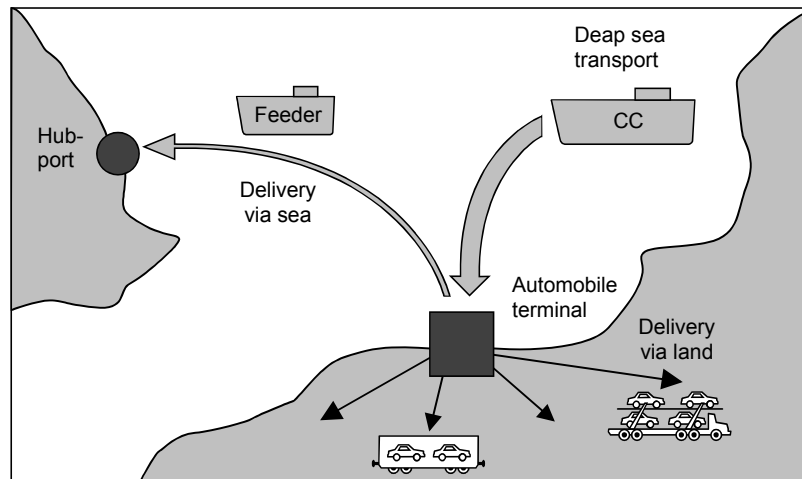


Fig.1: Transhipment process via a seaport automobile terminal

An overview of the transhipment process is given in Fig.1. The vehicles imported into Europe arrive by sea via a so called car carrier ship (CC). First, they are unloaded in the quay area, which has to be cleared in a relatively short time. Then the vehicles are stored in the terminal's parking areas. Starting from their actual position the vehicles are finally removed from storage to be delivered. The delivery can take place either by sea via a so called feeder ship or by land via truck or rail. Since the vehicles are moved self-propelling by drivers organised in gangs, the terminal operations are very cost-

intensive. The terminal's own drivers are employed on the basis of shifts. In the case of personal bottlenecks, additional drivers are hired by the shift or the hour from a port-wide workforce pool. The occupation of hired drivers leads, however, to an increase of transshipment costs per vehicle. An increase of cost is also caused by the storage of vehicles in the supplementary car park. In order to enhance the efficiency of the terminal operations, the handling of the vehicles is to be planned in such a way that the number of required drivers is minimised and balanced over the shifts in a given planning period.

The resulting vehicle transshipment problem is characterised by a high degree of complexity. Modern planning approaches like multi-agent systems seem therefore to be promising. Taking advantage of an approach by *Mattfeld and Kopfer (2001)*, the vehicle transshipment problem is first described as a planning problem comprising the subproblems quay management, storage allocation and deployment scheduling, *Fischer and Gehring (2001a,b, 2002)*. It can be completed with another subproblem, the estimation of the prospective departure time, *Fischer (2003)*. The quay management problem, which aims to pass unloaded vehicles through the quay area as quickly as possible, is an independent subproblem. In contrast, storage allocation, deployment scheduling and the estimation of the prospective departure time are closely interwoven. For this reason they form the focus of this paper, which will introduce a multi-agent system for the integrated handling of these subproblems. To make the focus clear, the handling of vehicles is described in more detail:

To simplify the planning of the vehicle transshipment, the vehicles are operated in groups. Every group consists of vehicles of the same or a similar type which are produced by the same manufacturer and are delivered by the same transportation facilities. Usually all relevant planning data are known beforehand. In exceptional cases, however, the departure time is not announced by car manufacturers up to the time of the delivery at the terminal. In these cases the prospective departure time has to be estimated to enable a timely planning. The handling of a vehicle group follows a sequence of tasks, the so-called task chain. After delivery, vehicles are unloaded into an incoming buffer store, taken from there into the terminal's parking areas, stored there temporarily, moved to an outgoing buffer store and finally loaded onto a carrier for transportation to the final destination. To facilitate the handling of the groups, each of the groups is stored in an interconnected part of one of the parking areas. If a sufficient free area is not available for a unloaded vehicle group, a trial is made to extend one of the free areas. An extension can be achieved e.g. by the relocation of a vehicle group stored in direct neighbourhood of this free area. If the trial fails, the vehicle group has to be stored in the supplementary car park, which has a less-well reinforced surface and is further away from the quays than the regular parking areas. Hence, there exist three types of task chains which can be run through by a group of vehicles:

- type 1: Unloading – insertion into storage – removal from storage – loading,
- type 2: Unloading – insertion into storage – relocation – removal from storage – loading,
- type 3: Unloading – insertion into supplementary car park – removal from supplementary car park – loading.

The execution of each task of a task chain is restricted by a time window. The time windows for tasks are limited by e.g. the docking and sailing times of ships, the arrival of other ships, the arrival of delivery carriers etc. The operation of vehicle groups along the task chains includes a series of decisions. The decisions aim at the selection of parking areas and buffers, the determination of the starting times of the tasks unloading, insertion, etc. and the determination of the number of drivers for the execution of these tasks. Additionally, in some cases prospective departure times have to be estimated in advance. With respect to the planning process the following details apply:

- The port buffer to be used for an unloaded group of vehicles is determined in the preceding quay plan. The outgoing buffer is fixed by the carrier or the customer.
- Every moving tasks for a group of vehicles has to be carried out by the assigned gang of drivers and completed within an eight-hour shift. For purposes of planning and control of terminal operations, the amount of work covered by a moving task is usually measured in so-called driver-

time cycles of a given length, e.g. fifteen minutes. A driver-time cycle is the amount of work which has to be invested to move one vehicle by one driver over a time span of fifteen minutes.

- For organisational reasons it is practical to carry out the task with gangs comprising constant numbers of drivers. Here, a gang consists of a minimum number of two drivers.

The problem of integrated storage allocation and deployment scheduling can now be formulated as follows, cf. *Fischer and Gehring (2001-2) and (2002)*: let be given a set I of vehicle groups i , $i \in I$, which have to be operated during a given planning period subdivided into eight-hour shifts. Let c_{1i} , $i \in I$, be the cost in driver time-cycles for operating the i -th vehicle group if task chain type 1 is executed, c_{2i} , $i \in I$, the corresponding cost in the case of task chain type 2 and c_{3i} , $i \in I$, the cost of task chain type 3. Let x_{1i} , x_{2i} and x_{3i} be binary decision variables which take the value 1 if the task chain type 1, type 2 and type 3 is executed for the i -th vehicle group and the value 0 otherwise. Since only one of the three task chains can be run through, the cost c_i of operating the i -th vehicle group is: $c_i = x_{1i} \cdot c_{1i} + x_{2i} \cdot c_{2i} + x_{3i} \cdot c_{3i}$, where $x_{1i} + x_{2i} + x_{3i} = 1$, $i \in I$.

Since the planning process aims at the minimisation and the balancing of the required drivers, the minimisation of the total number of driver-time cycles, TN , that have to be scheduled for transshipping the set I of vehicle groups in the planning period is used as the primary objective function:

$$\min TN = \sum_{i \in I} c_i \quad (1)$$

The secondary objective function is then the minimisation of the squares of the deviations, SD , between the maximum and the mean driver number over all shifts:

$$\min SD = \sum_{t=1}^T (\hat{f}_t - \bar{f}_t)^2 \quad (2)$$

Here T denotes the number of successive eight-hour shifts t , $t = 1, \dots, T$, in the planning period, while \hat{f}_t denotes the maximum and \bar{f}_t the mean number of drivers deployed in the t -th shift.

The rest of the paper is organised as follows: in section 2 a multi-agent system supporting the integrated storage allocation and deployment scheduling is proposed. In section 3 the multi-agent system is evaluated using test instances, which were generated in conformity with practical conditions. Finally, some conclusions are drawn in section 4.

2. A multi-agent system for supporting the operational planning tasks

Under practical conditions the subproblems of departure time estimation, storage allocation and deployment scheduling are usually handled by different decision-makers. Considering the organizational structure of a terminal, an analogous distribution of decision competences can be used in a decision support system for the integrated planning problem considered here. In recent years multi-agent systems (MAS) have been recommended for supporting complex decision-making problems that can be decomposed into interdependent subproblems, *Brenner et al. (1998)*, *Jennings et al. (2000)*. According to the literature, MAS are applied to areas ranging from production and transportation planning, *Zelewski (1998)*, *Fischer et al. (1996)*, through to business process management, *Jennings et al. (2000)*. A multi-agent based approach for supporting the vehicle transshipment problem has not yet been presented in the literature. The MAS proposed here is described in two steps, the first step concerns the architecture and the second the operation of the MAS.

A distribution of decision competences as mentioned above is achieved by the use of a specific type of agent for each of the problems of departure time estimation, storage allocation and deployment scheduling. For the departure time estimation the so-called departure time estimator agent (DEA) is used and for the storage allocation to the so-called area agent import (AAI). In contrast to the departure time estimation and the storage allocation, the shift-based deployment scheduling requires a decentralised decision-making over the time. For this reason, deployment scheduling is carried out by a number of S so-called shift agents ($SA_s, s = 1, \dots, S$). Since a shift agent always takes over the planning for exactly one shift, only a number of $n < N$ of shift agents is concurrently active. A further agent, the so-called planning coordinator agent (PCA), is responsible for the coordination of the local planning of the individual agents. The overall architecture of the MAS is shown in Fig.2.

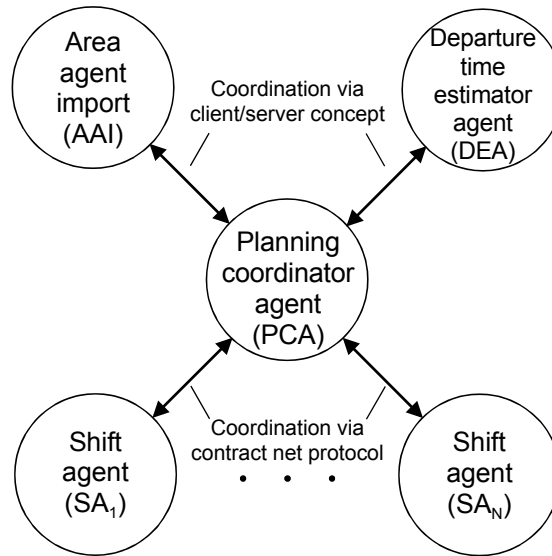


Fig.2: Architecture of the multi-agent system

The planning competences of the DEA respective AAI are rather limited. Therefore, the coordination between PCA and these agents is reduced to a type of client/server relation. If the departure time for a vehicle group is unknown, PCA submits the respective estimation problem to the DEA, who estimates the departure time and returns the solution back to PCA. The co-ordination between the PCA and the AAI is performed in the same way except that AAI has to solve a storage allocation problem for a vehicle group. A different situation is given for the deployment scheduling for a vehicle group, because each task of the associated task chain can often be carried out in different shifts. The problem of selecting a shift agent can be solved by e.g. an auction mechanism based on the contract net protocol, *Sandholm 1998*). Here a contract network protocol is used which integrates the coordinator variant model introduced by *Zelewski (1993)*. The purpose of this model is the elimination of the coordination deficiencies that occur with decentralised forms of coordination.

The design of the architecture shown in Fig.2 considers the three typical features of a MAS, *Fischer and Gehring (2002)*. First, the decentralised structure of the underlying planning problem is modelled in conformity with reality. Second, the introduced agents are autonomous and pursue different goals. Third, a mechanism for coordinating the local planning processes is used; this mechanism is embedded in the PCA. The conceptual and procedural realisation of these features is specified in the following.

The operation of the MAS starts with some initialisations and ends with some evaluations performed by the PCA. In between five activity phases of the MAS can be distinguished, which are executed until all vehicle groups have been processed. In detail these are:

- estimation of the departure time for a vehicle group (if necessary),
- deployment scheduling (coarse-grained scheduling) for a vehicle group,

- storage allocation for a vehicle group,
- deployment scheduling (fine-grained scheduling) for each task to be executed for a vehicle group and
- updating of the so-called parking area reservation list (if necessary).

In the first phase the *estimation of departure time* is carried out by the DEA, if the departure time of a vehicle group is not even known at the date of their arrival at the terminal. In this case, the DEA estimates the prospective departure time on the basis of his own experience values, which can be adapted to their appearance over the whole planning period. The DEA also corrects estimated departure times finally as announced by the car manufacturers.

In the second phase the *deployment scheduling* (coarse-grained scheduling) for a vehicle group is performed on the level of individual tasks. For this purpose the three interaction steps of the applied contract net protocol are executed for each task. These steps are: invitation to bid, submission of offer and order. As to the invitation to bid, the PCA first determines the shifts being considered for the execution of the individual task. The respective shift agents are then supplied with the commission data. In the second step each of the contacted shift agents returns an offer to the PCA. In the third step the PCA selects the most favourable shift for carrying out the individual task. If the task time window overlaps with one shift only, then this shift is selected. Otherwise, all offers are evaluated by means of a lexicographical function. The shift is selected which is, in decreasing order, characterized by the longest overlapping period, the largest number of free driver-time cycles for regular drivers, the largest number of free driver-time cycles for hired drivers, the smallest number driver-time cycles for hired drivers and the lowest shift index.

In the third phase the *storage allocation* for a vehicle group consists of two steps. In the first step, the PCA submits data on the respective vehicle group to the AAI. On the basis of these data the AAI determines a parking area for the vehicle group and returns his decision to the PCA. To determine a parking area, the AAI has the option of two algorithms: a simple heuristic procedure (HP) or a genetic algorithm "multi-sequence gene" (MSG) (cf. *Fischer and Gehring 2001-2 and 2002*).

The central concept used by the heuristic HP is a "multidimensional priority list" consisting of several sublists. For each combination of a port buffer at the quay side with an outgoing buffer at the delivery side a corresponding sublist is included in the multidimensional priority list. A sublist contains the corresponding de facto transportation times for a vehicle in minutes for all transportation paths connecting a given port buffer via one of the regular parking areas with a given outgoing buffer. Each entry in a sublist consists of a parking area identifier and a time coefficient. The entries are sorted with increasing order of the coefficient values.

The multidimensional priority list is used to determine a parking area for a vehicle group. First, an attempt is made to place the vehicle group in a single parking area. If one or more suitable parking areas are available, the area with the highest priority, i.e. the lowest time coefficient is selected (type 1 task chain). Otherwise, an attempt is made to enlarge a free area by relocating vehicles according to the type 2 task chain. If this attempt fails, then the vehicle group is stored in the supplementary car park (type 3 task chain).

The problem representation used by the genetic algorithm MSG follows the concept of the parking area priority list. This means that a chromosome represents a multidimensional priority list which consists of several sublists. Each sublist consists of a permutation vector of parking area identifiers which corresponds to a combination of a port buffer with an outgoing buffer. The decoding procedure is similar to the heuristic HP, however, in contrast to HP a multidimensional priority list is used and relocations of vehicles are excluded. The representation used by MSG is therefore closer to reality.

In the fourth phase the *deployment scheduling* (fine-grained scheduling) for a vehicle group is also performed on the level of individual tasks. For each task of the respective task chain the PCA submits

the order for planning the task to the shift agent of the selected shift. The shift agent executes the order in three steps: First the operation time of the task which leads to the lowest loss of time is determined. A loss of time occurs if the operation time of the task, which is given in whole time cycles of 15 minutes, is greater than the actual driving time measured in minutes. Time losses may also occur in situations where each driver moves more than one vehicle. For the determined operation time, the starting time of the task is then fixed, so that the new local load peak for the drivers caused by the task is low as possible. Finally, the task is scheduled taking account of the starting time determined in the second step.

In the fifth phase the *parking area reservation list is updated*. For each regular parking area this list contains the number of parking areas that are still free, the partial space reservation for vehicle groups and the start and end of reservation. The parking area reservation list is managed by the AAI and must always be updated if a vehicle group is stored on a regular parking area, i.e., a type 1 or type 2 task chain is carried out. In these cases the PCA submits a corresponding order to the AAI. The parking area reservation list is used by both of the storage allocation algorithms, HP and MSG.

3. Evaluation of the multi-agent based planning approach

For the implementation of the MAS the language Java was used. The MAS was tested on a Pentium IV PC with a cycle frequency of 2.0 GHz. In the following, the test concept and the numerical results derived by series of test runs are described.

For the test a total number of 15 problem instances were generated. All instances are based on the same locations of parking areas. In conformity with the situation in an European seaport the following facilities are assumed: two port buffers for 750 and 1000 vehicles respectively, ten standard parking areas with a capacity of 7000, 6000, 5000, 5000, 4000, 4000, 3000, 3000, 2000, and 1000 vehicles, six outgoing buffers with a capacity of 1000, 1000, 750, 500, 500, and 500 vehicles, and a supplementary car park with sufficient capacity. For the generation of problem instances a planning period of 750 shifts was chosen. According to the practical situation in the mentioned seaport, a total of 400000 vehicles are imported within that period of time. In accordance with the manufacturer, the lengths of stay of the vehicles in the terminal are varied over the problem instances. For every 5 instances a defined length of stay is used. While the average length of stay varies between 100 and 200 shifts, the spread of the averages lies in a range of approximately 10% to 20%. The seed value z of the random number generator used was also varied. The seed values lie in a range 100 to 500 and for every 3 instances the seed values 100, 200, 300, 400, and 500 are used.

On the basis of these problem instances two series of numerical experiments were carried out, each considering a specific planning situation:

- From the operational point of view of the responsible operations manager, the values of the global objective functions that can be achieved through the application of the MAS and the embedded allocations methods HP and MSG are of main interest.
- From a more strategic point of view, the question of how many drivers should permanently be employed is in the foreground.

In the following, the operational and the more strategic aspects of the application of the MAS are considered.

As to the operational aspect, each of the 15 problem instances was calculated once using the heuristic procedure HP, and the genetic algorithm MSG. The objective function values that were achieved, measured in quarter-hour time cycles, were always averaged over all 15 instances (see Table I). In order to examine the influence of different relations between regular and hired drivers on the solution behaviour of the MAS, different numbers of regular drivers were introduced.

Table I: Averaged objective function values for solution methods HP and MSG.

No. of regular drivers	Solution method	Averaged objective function values	
		Mean total numbers of driver time-cycles (objective function (1))	Mean squares of deviations (objective function (2))
20	HP	1.944.626	3.502.712
	MSG	1.778.445	3.297.325
30	HP	1.967.357	3.373.498
	MSG	1.788.980	3.259.144
40	HP	1.984.200	3.295.305
	MSG	1.783.064	3.148.428
50	HP	2.005.530	3.271.621
	MSG	1.792.501	3.035.176
60	HP	2.007.154	3.143.381
	MSG	1.797.560	2.978.934
70	HP	2.009.307	3.100.632
	MSG	1.802.058	2.907.524
80	HP	1.997.427	3.002.486
	MSG	1.788.750	2.878.044
90	HP	1.980.744	2.959.673
	MSG	1.765.243	2.837.384
100	HP	1.957.906	2.887.345
	MSG	1.751.267	2.801.479

From Table I it can be depicted that the genetic algorithm MSG dominates clearly over the heuristic HP for all numbers of regular drivers. The average improvement in the values of the objective function (1) achieved with MSG in comparison with HP ranges from approximately 4.15% for 20 and approximately 6.46% for 100 regular drivers. Similar improvements were calculated for the objective function (2). For the heuristic HP the average computing time per problem instance amounts to 15 seconds; the corresponding value for MSG amounts to 1450 seconds. These computing times were, however, derived for computational experiments covering 750 shifts or one year and are therefore of no practical relevance. Orders are usually known between one and six weeks beforehand. This means that the MAS could be applied, e.g., every two or three days with the consequence that computing times would be reduced dramatically. The MAS seems therefore to be a suitable tool for supporting the continuous planning of the current terminal operations.

From a more strategic point of view, the question of how the total personnel costs can be influenced is of high interest. The reason is that transshipment per seaport terminals is particularly labour-intensive and leads to high personnel costs. The level of the total personnel costs is mainly influenced by regular drivers and by the surcharge rate for hired drivers. In order to show the influence of these critical factors, the sum of the relative costs of regular and hired drivers was calculated for different numbers of regular drivers and for different surcharge rates. The numerical results obtained are shown in Fig.3. Idle times of regular drivers are relevant now. Therefore, the relative costs of regular drivers are proportional to their number. The relative costs of hired drivers were determined as the product of the corresponding mean total numbers of driver-time cycles (including idle times of hired drivers) and the cost factors 1.25, 1.5, 1.75, and 2.0 for a surcharge rate of 25%, 50%, 75%, and 100%.

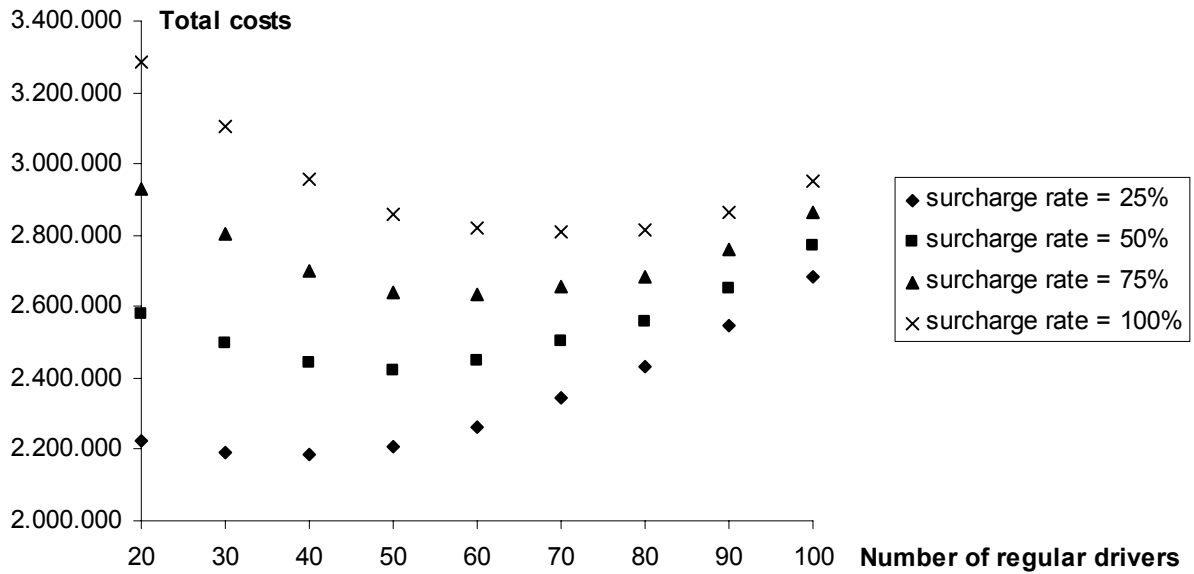


Fig.3: Total cost curves for different numbers of regular drivers and for different surcharge rates

As Fig.3 shows, there exists always a cost optimum. The position of the optimum depends on the level of the cost surcharge per hired driver. With the increase of the surcharge rate from 25% to 50%, 75%, and 100%, the position of the cost optimum is shifted from about 40 to about 50, 60, and 70 regular drivers. This means that an increase in the costs of hired drivers can be compensated by extending the number of regular drivers.

The optimisation potential indicated by Fig.3 is of practical relevance. If the number of regular drivers is varied in the given range, then the difference between the minimum and the maximum total costs amounts to 18%, 13%, 11%, and 16% for a surcharge rate of 25%, 50%, 75% and 100%. These economic dimensions seem to justify the multi-agent based decision support approach presented here from a more strategic point of view, too.

4. Conclusions

The planning of vehicle transshipment of imported finished vehicles via a seaport automobile terminal includes two planning tasks: the allocation of parking areas for the temporary storage of vehicles and the allocation of drivers to vehicles that have to be moved in the terminal area. In addition, it may be necessary to estimate the delivery times of vehicles. These planning tasks are assigned to three different agent types of a multi-agent system (MAS). A further agent, the coordinator agent, coordinates the activities of other agents in such a way that the demand for drivers in the planning period is minimised and balanced.

The multi-agent system was tested using 15 problem instances, which were generated in conformity with the situation in an European seaport. The test aimed at evaluating the MAS from an operational and, in addition, a more strategic point of view. The test led to the following results: The MAS seems to be a suitable tool for supporting the continuous planning of terminal operations. Further, the planning results can be improved significantly if a genetic algorithm instead of a heuristic is used for the allocation of parking areas. As to the strategic aspect, the total relative costs for regular and hired drivers have an economically relevant minimum, which is shifted towards a higher number of regular drivers for an increasing surcharge rate. Hence, the MAS seems also to be suitable to determine a stock of permanently employed drivers.

References

- BRENNER, W.; ZARNEKOW, R.; WITTIG, H. (1998), *Intelligente Softwareagenten – Grundlagen und Anwendungen*, Springer
- FISCHER, K.; MÜLLER, J.P.; PISCHEL, M. (1996), *Cooperative transportation scheduling: an application domain for DAI*, J. Applied Artificial Intelligence, Special Issue on Intelligent Agents 10/1, pp.1-33
- FISCHER, T. (2003): *Ein Learning-Classifer-System zur Schätzung von Lagerverweildauern im Rahmen des Fahrzeugimporte*, In: Bortfeldt, A.; Fischer, T.; Homberger, J.; Pankratz, G.; Strangmeier, R.: Planen, Lernen, Optimieren – Beiträge zu Logistik und E-Learning. Diskussionsbeitrag Nr. 338 des Fachbereichs Wirtschaftswissenschaft, FernUniversität in Hagen, Mai 2003, pp.37–54
- FISCHER, T.; GEHRING, H. (2001a), *Ein Multi-Agenten-Ansatz zur Lösung des Fahrzeugumschlagproblems*, In: Fleischmann B. (ed.): Operations Research Proceedings 2000. Springer, pp.401-406
- FISCHER, T.; GEHRING, H. (2001b), *Ein genetischer Algorithmus zur Stellflächenplanung im Rahmen des Fahrzeugumschlagproblems*, In: Sebastian, H.-J. and T. Grünert (ed.): Logistik Management, Teubner-Verlag, Berlin/Heidelberg 2001, p. 365-376.
- FISCHER, T.; GEHRING, H. (2002), *Planung des Fahrzeugumschlags in einem Seehafen-Automobilterminal mittels eines Multi-Agenten-Systems*, FernUniversität Hagen, Fachbereich Wirtschaftswissenschaften, Diskussionsbeitrag Nr. 323, September 2002.
- JENNINGS, N.R.; FARATIN, P.; NORMAN, T.J.; O'BRIEN, P.; ODGERS, B. (2000), *Autonomous agents for business process management*, Int. J. Applied Artificial Intelligence 14/2, pp.145-189
- MATTFELD, D.; KOPFER, H. (2001): *Terminal Operations Management in Vehicle Transhipment*, Universität Bremen, Lehrstuhl für Logistik, Technical Report
- SANDHOLM, T. (1998), *Contract Types for Satisfying Task Allocation: Theoretical Results*, AAAI-98 Spring Symposium: Satisficing Models, Stanford University
- ZELEWSKI, S. (1993): *Multi-Agenten-Systeme für die Prozesskoordinierung in komplexen Produktionssystemen*, Universität zu Köln, Seminar für allgemeine Betriebswirtschaft, Industriebetriebslehre und Produktionswirtschaft, Arbeitsbericht Nr. 46
- ZELEWSKI, S. (1998), *Multi-Agenten-Systeme – ein neuartiger Ansatz zur dezentralen Feinterminierung von Produktionsaufträgen*, Wildemann, H. (ed.): Innovationen in der Produktionswirtschaft. Transfer-Centrum-Verlag, München 1998, p. 133-166.

Visualisation of Ship Motions in Waves and during Grounding for Simulators

Meelis Mäesalu, Helsinki University of Technology, Espoo/Finland, Meelis.Maesalu@hut.fi
Jerzy Matusiak, Helsinki University of Technology, Espoo/Finland, Jerzy.Matusiak@hut.fi

Abstract

A system that enables 3D-visualisation of ship motions in waves and ship grounding is developed. The architecture of such a visualisation system is presented. The system comprises three parts: The first part is the entity transferring the ship data provided by the NAPA ship design software into the form used by the software evaluating ship dynamics and also utilised by the visualisation system. The second part, called LAIDYN, evaluates ship dynamic responses. In this method, the ship is regarded as an intact rigid body. Large amplitude non-linear ship motion in waves or ship dynamic response due to grounding is simulated. The third part, which is described in greater detail, is the visualisation. The basic entities and tools for producing 3D visualisation are presented. The programming language Visual C++ 6.0 and the GLUT visualisation library are used. The objects necessary for the visualisation of ship motion and grounding are defined and their properties are discussed. The user interface for the visualisation system is presented. Also, the implementation of the entire system is described. Two examples of the visualisation system application are presented. The first one deals with parametric roll resonance of a Ro-Ro vessel in following seas and the other visualises a hard power grounding of another Ro-Ro ship. The possibilities for further development of the system for a ship-handling simulator are discussed.

1. Introduction

Computer aided 3D visualisation offers new possibilities to analyse a variety of problems in ship design by transforming complicated ship data into a computer simulation. In particular 3D-visualisation of ship motions is a valuable tool in ship design. A system that enables visualisation of ship motions in waves and motions due to ship grounding is developed and presented in the following.

2. General Architecture of Ship Motion and Grounding Visualisation System

Accurate scientific visualisation of large amplitude non-linear ship motion in waves or during a grounding accident is based on complex data of the ship structure, and simulated motions in waves or due to the grounding of the ship. The general architecture of the data flow for such a system is presented in Fig.1. The visualisation process is divided into several steps. The first step is the collection of the relevant ship data. The next step is creation of the ship's panelised hull form and calculation of other relevant particulars. The input data is calculated by the Naval Architecture package computer program NAPA. The next step is the calculation of ship motions in 6DOF. In the final step the data has to be read into the visualisation system, which then produces an accurate 3D visualisation of the ship's motions before, during and after the accident.

3. Input data collection

3.1. Ship hull surface generation and primary particulars

The ship hull geometry is generated by defining some primary curves of the ship hull surface NAPA (1999). Normally, the ship hull surface is divided into three parts, which are aft-, mid- and forebody. The primary curves for the forebody are the stem, deck, section, flat bottom and -side, and knuckle curves. If a ship has a bulbous bow, then a minimum of two curves need to be defined, one curve for the bulb geometry description and the other for describing the bulb connection with the bow. For the midship, these main curves are sections. Of the centre, flat bottom and -side curves, at least two must be defined. For the aftbody the primary curves are the deck, section, transom, stern, flat bottom and -

side curves. If the ship has no parallel midbody, then the aft- and forebody only are used for the ship hull description.

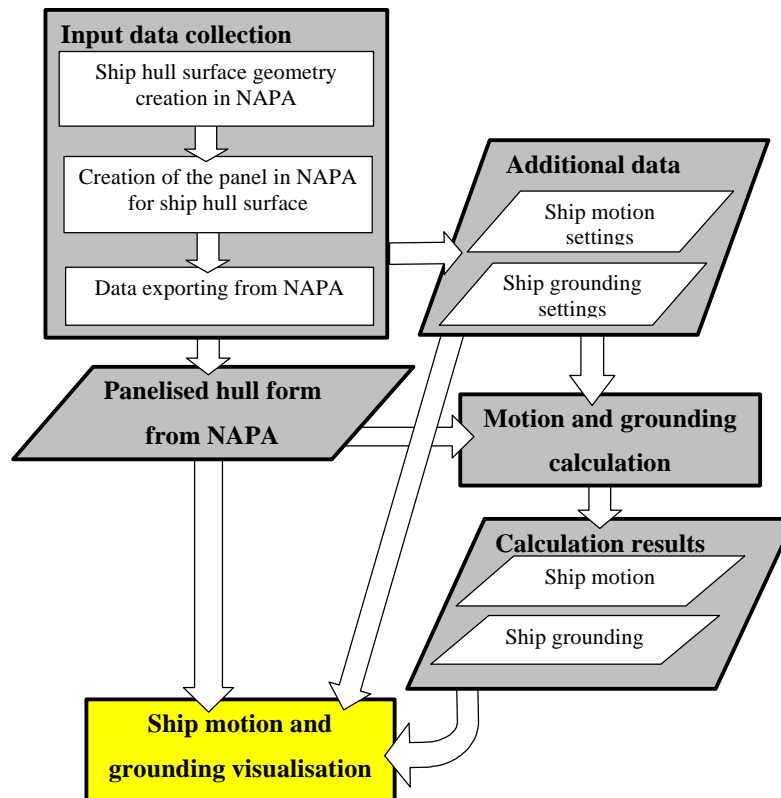


Fig.1: General architecture of the data flow for visualisation system

Based on the primary curves, the NAPA software generates the surface geometry of the fore-, mid- and aftbody. Then the parts are connected with each other and the program calculates the primary particulars of the ship, such as, displacement of the ship in actual draft, centre of buoyancy, etc.

3.2. Geometry transformation

The created ship hull form is not compatible with the input data of the other programs. Therefore the geometry has to be transferred to a form which is compatible with the other programs. For the surface geometry transformation different possibilities exist. The *iges* format is largely used in the industry as a format for surface geometry transformation. For lines the *dxf* format transformation is most widely used. In the scientific and virtual environment (VE) field the *iges* format is not often accepted, because in this format it is not possible to add any texture. Additionally, the polygon amount control is not simple. In this field the most importance formats are *3ds*, *flt* and *wrl* format. A complicated geometry transformation with a small number of polygons generally yields problems. In the industry, during manufacturing, the accuracy is very important and then the amount of polygons doesn't matter. But in the scientific and VE field selection of the appropriate amount of polygons is very important in order to produce a good simulation and visualisation.

In the current case, not only the surface geometry transformation is needed, but also the so-called metadata associated with the geometry has to be transformed. Therefore some other method was needed. One possibility to transfer this information is to discretise the hull form using a number of panels. This was chosen as the best method for data transfer for the programs which calculate the ship motion and grounding. Later this method was used also for the visualisation. A panel is by definition a four-sided polygon, and accordingly panelisation means in this context that the ship hull form is approximated with panels. Approximation accuracy depends on the amount of panels used.

3.3. Panelisation and information exporting

The NAPA program has an auxiliary subsystem called NPN for surface object panelisation. Panelisation is done by using the macro language of NAPA. The panel mesh representation of the ship hull can be seen in Fig.2.

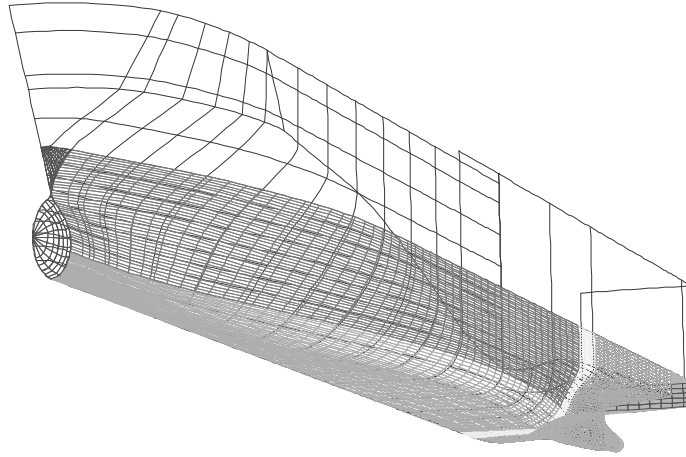


Fig.2: Panel mesh representation of the ship hull (4870 panels are used to describe half of the ship)

Fig.3 shows how panels are defined for the processes. Points 1, 2, 3, 4, are the panel corners. Point C is the panel centre point, at which the program calculates the normal vector. S_n is the area of the current panel. This means totally 16 numbers are needed to describe a single panel. For exporting that information the macro language is used.

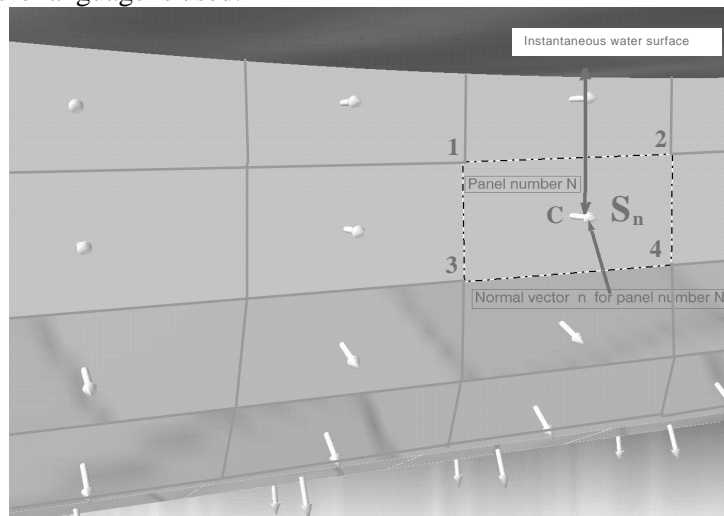


Fig.3: Panel data exported to other applications

4. Dynamics of Ship Motion

A general model of ship dynamics incorporating non-linear seakeeping in six degrees-of-freedom and manoeuvring is used *Matusiak (2002)*. There are no restrictions set on the motion's magnitude. The ship is regarded as a rigid body. The actions of the propeller and rudder are taken into account by relatively simple semi-empirical models. An allowance for a podded propulsor is made. Motion calculations are conducted in the time domain.

4.1. Ship Motions in Waves

Ship motions in waves are simulated using the so-called two-stage approach *Matusiak (2002)*. In this approach the general model of ship dynamics in waves is divided into the linear part and the non-linear portion. The linear ship response in waves is evaluated using a linear strip theory. The non-linear portion comprises the cross-coupling terms of body dynamics and non-linear parts of the restoring and Froude-Krylov forces. The restoring and Froude-Krylov forces are evaluated with the ship hull represented by a three-dimensional panel model. Radiation and diffraction forces are assumed to be sufficiently well represented by the linear model. Radiation forces in the time domain are obtained making use of the so-called retardation function formulation. A set of 13 non-linear ordinary differential equations is solved numerically in the time domain yielding the ship forward motion and the non-linear portion of the other responses. The 13th equation yields the rudder angle for a ship steered by an autopilot. As a result, the total response of the ship in terms of three displacements in the inertial co-ordinate system and Euler angles is obtained.

4.2. Ship Grounding Calculation

It is assumed that the ship is intact. This means a possible flooding due to grounding is not taken into account. A single force vector represents grounding. At each time step of the simulation, the minimum distance between the control points representing the hull and rock tip is sought. This distance determines whether the contact occurs and it gives the panel number at which contact occurs as well as the penetration. Moreover, the direction of the relative velocity between the panel and the rock tip is determined. If contact occurs and the normal component of the relative velocity points inside the hull, the contact force is evaluated. Both the normal and the tangential component of the contact force are assumed to be simple functions of penetration. When simulating the model experiments, the constitutive relation of the force and the penetration is used as obtained experimentally by *Lax (2001)*. In these tests the bottom of the ship model was fully constructed of plastic (inelastic foam) which when scaled to the ship's scale gave realistic values of the contact force as a function of the penetration. Also a more realistic relation obtained with an aid of structural analysis can be used *Kajaste-Rudnitski (2003)* for the full-scale simulations.

5. Visualisation

Visualisation means making things visible *Mantere (2001)*. Visual material like graphs or 3D animations are easier understood than, e.g., equations or data in tables. Even more so, visualisations often help us to see things in a new way. Therefore, a lot of visual techniques are being developed. The purpose of visualisation is not to produce a nice picture, but to stress the essential part of a data set. As it is said about computers, "the purpose of computation is insight, not pictures". Thus, the ultimate task of scientific visualisation, with information visualisation as subtask, is the exploration of phenomena and information sharing between professionals of different fields, in a word, insight *Card et. al. (1999)*.

5.1. Scientific visualisation

Fig.4 shows a reference model for the visualisation of scientific data and areas where humans can interact with the visualisation procedure. Usually, there is a lot of raw data that first must be ordered, filtered or transformed, so only relevant data is visualised. First, for a formal data visualisation the relation between original data and metadata has to be selected from the data table. Data from natural sources often have explicit spatial relations and their use is obvious. In the next phase, visible structures and objects are generated from the data tables. This is the most important phase in the visualisation procedure, because the data is transformed into visual form. In the last phase, the visual output is given a convenient form, which demands specification of, e.g., position in space, scale and clipping plane.

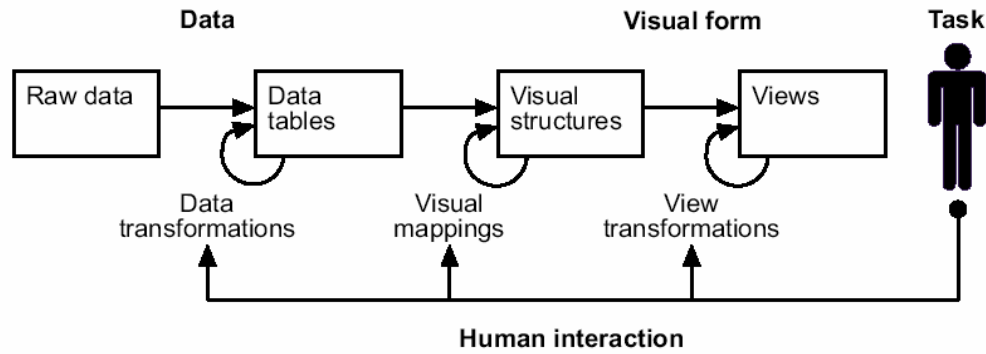


Fig.4: Reference model for data visualisation, *Mantere (2001)*

5.2. 3D visualisation

The power of 3D visualisation is based on its spatial characteristics. Frequently, 3D visualisation is used for the illustration of physical problems of spatial nature. Two main methods are used to implement the visualisation, one being based on surface graphics, the other on volume graphics. Surface graphics use vector calculations for defining points, lines and surfaces as geometric elements in an empty space. Volume graphics divide the space into small cubical volume units, and, for every unit, one or more values are defined, *Mantere (2001)*. In this work, the surface graphics technique is used.

In 3D visualisation it is very important to set the viewpoint properly, otherwise essential information is hidden behind other objects. Therefore, primary operations such as viewpoint changes, scaling and clipping have to be interactive processes. Also intentional enhancement can be used to highlight a specific area of the data. E.g., intentional distortion is largely used in FEM results analysis to better visualise small displacements. The basic entities and tools used in our visualisation are pixel, polygon, vertex, light sources and illumination, shading, texturing, bounding box, clipping plane, z-buffering and level of detail. Virtual environment construction is a mathematically complicated process, where every pixel needs to be calculated several times before the total geometry can be calculated and projected. Today, polygon use is accepted by the 3D society as the standard for 3D representation. For that reason most graphic cards are optimised for polygon filling with different effects. Thus the processing load for the main computer processor is minimised, because rendering uses the graphics card and the main processor is needed only for geometry calculations and simulations.

For routine performances many different 3D libraries and toolkits are developed. E.g., libraries can contain different filling routines. The most popular visualisation libraries are OpenGL, DirectX, GLUT, World ToolKit and Open Scene Graph. The two first of them are low level visualisation libraries and, accordingly, the others high level visualisation libraries. Lots of others 3D libraries exist, but their use is not commonly accepted.

5.3. Used platform and software

The ship motion and grounding accident visualisation system was developed using a Visual C++ 6.0 compiler and GLUT-visualisation libraries. These were chosen for several reasons. The first and most important criteria were that GLUT contains all primary components necessary for developing a visualisation system. Additionally, it is compatible with the Windows NT operating system and freely distributed without licence fees. All the primary objects in the visualisation system have to be created dynamically by polygons, therefore the use of more sophisticated platforms is not necessary.

6. Implementation of the visualisation system

The developed visualisation system does not work in real time. One reason for this is that the calculation of ship motions is too slow. This is the reason why the calculations have to be conducted

beforehand. The other reason is that the visualisation itself needs a powerful computer. But the aim of this visualisation system was to develop a toolkit which allows the examination of ship motions in any time-step independently of the computer performance.

The visualisation system contains many co-ordinate systems, since each object is first created using its own object-fixed co-ordinate system. Each of them is later on set into spatial relationship to the whole scene and thus transferred to the so-called global earth-fixed co-ordinate system used for visualisation.

Ship motion is described by using several co-ordinate systems, Fig.5, all being right-handed. The NAPA co-ordinate system ($X_N Y_N Z_N$) is necessary for ship hull generation; its origin is located at point N. This NAPA co-ordinate system is not useful for describing ship motion, which needs a body-fixed co-ordinate system (xyz) with the origin at the centre of the gravity of the ship, point G, Fig.5. The origin of the third co-ordinate system ($X_G Y_G Z_G$) coincides with the second (xyz) one, but this co-ordinate system has the same orientation as the earth-fixed co-ordinate system. The fourth co-ordinate system is the earth-fixed co-ordinate system (XYZ). This last co-ordinate system is the co-ordinate system used for visualisation with the origin located at point L. This point L is the apex of the conical rock.

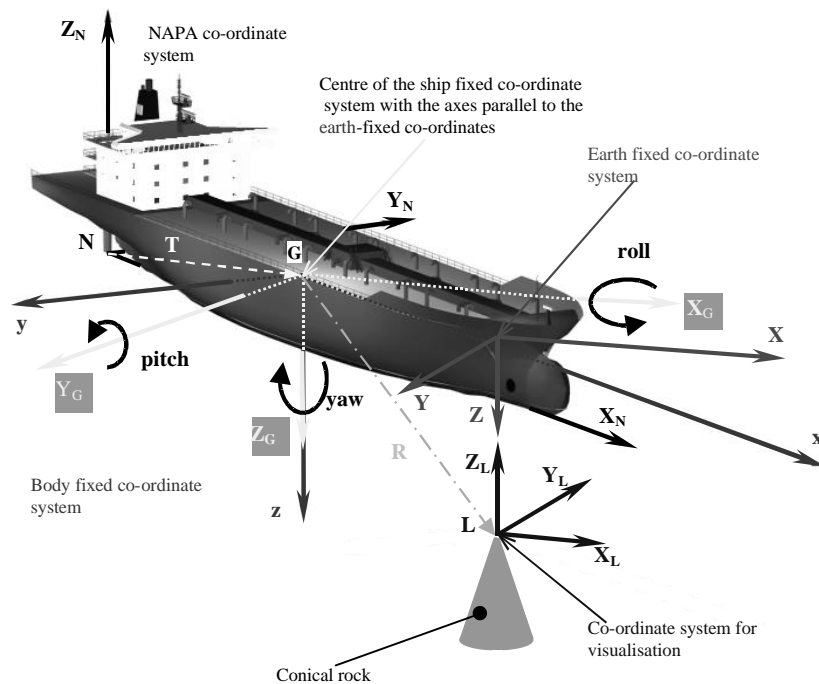


Fig.5: Ship motion describing co-ordinate systems used in the visual system

6.1. The User Interface

The user interface of a visualisation system is to be user friendly and as simple as possible. Therefore the navigation in the scene is done by using the keyboard. The menus offer general options and are chosen with the mouse, Fig.6.



Fig.6: Screenshot of the user interface

6.2. Examples of the visualisations

In the following, two examples of the visualisations are presented. The first one is visualisation of the grounding and the other one is ship motion in waves. In both cases model tests were made to verify the calculation routines.

6.2.1. Ro-Ro Ship Grounding Visualisation

Fig.2 shows the panel mesh of the investigated ship, which had the following main particulars: Length of the ship 146.27 [m], Breadth of the ship 25.35 [m], Draft of the ship 7.35 [m], Centre of gravity measured from the baseplane 10.0 [m]. Other navigational parameters are presented in *Matusiak and Varsta (2002)*. The model of the ro-ro ship was tested with a collision with a conical obstacle. The bottom of the model was covered with a soft, inelastic urethane material. Grounding forces were measured at the cone tip. For additional details of the experiment see *Lax (2001)*. The test parameters are the following:

Ship initial speed is $6 \left[\frac{m}{s} \right]$.

Depth of the rock tip, from still water level is 6.76 [m].

The ship centre of gravity has the following co-ordinates in the global co-ordinate system (origin of the co-ordinate systems coincide with the rock) before the simulation:

$X = -45$ [m]

$Y = -5$ [m] (from the ship's centre plane to port)

$Z = 9.41$ [m].

The calculation results are presented in the publications *Kajaste-Rudnitski (2003)*. Fig.7 shows the ship before, during, and after the grounding event.

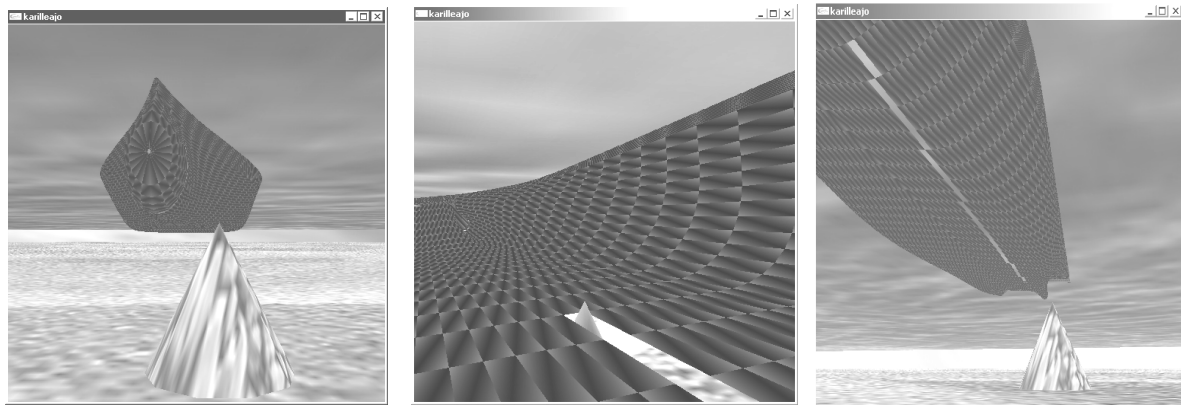


Fig.7. Ship grounding before, during and after grounding accident

6.2.2. Ro-Ro vessel rolling in head regular waves

The second case is another ro-ro vessel experiencing so-called parametric roll resonance in head and following waves. Details of the simulations and validation model tests are given in *Matusiak (2003)*. Fig.8 shows the result of the simulation and photographs taken during the model test experiments.

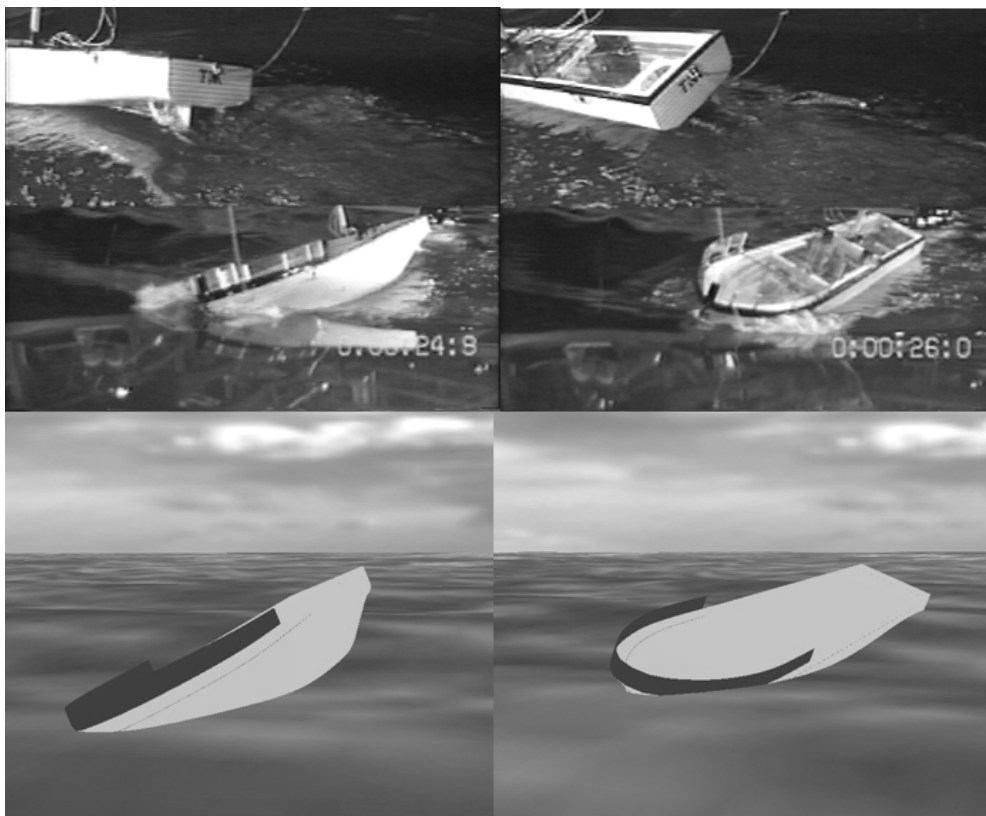


Fig.8. Ro-Ro vessel is rolling in head regular waves (model tests (top) and visualisation based on the numerical simulation (bottom)).

Visualisation of simulations compares well with the experiment. Although both the visualisation and the simulation do not include the diffracted waves, it helps to identify the critical events such as e.g. propeller emergence or water deck submergence. It also helps to identify the mechanisms leading to a loss of dynamic stability and capsizing of the vessel. The visualisation tool allows easy inspection of ship motion at any time step.

7. Conclusions

The presented toolkit enables any user to view ship behaviour in rough seas or during grounding. These particular problems in ship dynamics become clear with the aid of visual simulation. For extending this visualisation system and gaining more flexibility and speed, the relevant data for ship motion and grounding, as well as the environmental properties, should be saved in a database where they can be changed easily. Some modifications are needed to make the code work in real time. After that, this visualisation system can be used as part of a real time simulator. For use in a simulator, other already widely accepted graphics format readers also should be included. These would make it easier to read additional graphic elements such as e.g. ship superstructures. This will improve significantly the visualisation quality. These graphics formats can be *3ds*, *dxf* and *flt* formats. The system was, nevertheless, implemented in a way that easily allows these changes.

If the computer capacity increases sufficiently to allow the LAIDYN calculation routine to work in real time, then this part is possible to use as the simulator core. Autopilot steering can be easily changed to joystick steering. As a result, a real-time simulator of ship motions in 6 DOF can be constructed.

References

- CARD, S.K.; MACKINLAY, J.D., SHNEIDERMAN, B., (1999), *Readings in information visualization, using vision to think*, Morgan Kaufmann Publ.
- KAJASTE-RUDNITSKI, J.; VARSTA, P. (2003), *Mechanics of ship grounding*, HUT Ship Laboratory, M series report 282, Espoo
- MANTERE, M., (2001), *Visualization of flow data in photo-realistic virtual environment*, Master's Thesis, Helsinki University of Technology, Espoo
- MATUSIAK, J; (2002), *Two-stage approach to determination of large amplitude motions of a rigid ship in waves*, 15th Nordic Seminar on Computational Mechanics, Aalborg
- MATUSIAK, J.; VARSTA, P. (2002), *transient motion of ship during hard grounding*, 6th Int. Ship Stability Workshop, Glen Cove
- MATUSIAK, J., (2003), *On the effects of wave amplitude, damping and initial conditions on the parametric roll resonance*, 8th Int. Conf. Stability of Ships and Ocean Vehicles, Madrid, pp.341-348
- NAPA OY, (1999), *Introduction to NAPA*, 2nd Edition, Based on release 99.1, Helsinki

Innovative Robotic Solutions for the Survey and Certification of Ships and Mobile Offshore Units

Peter Weiss, Cybernetix, Marseille/France peter.weiss@cybernetix.fr

Fivos Andritsos, JRC, Ispra/Italy fivos.andritsos@jrc.it

Frédéric Schom, Cybernetix, Marseille/France frederic.schom@cybernetix.fr

Alain Fidani, Cybernetix, Marseille/France alain.fidani@cybernetix.fr

Abstract

Accidents on offshore oil production facilities and oil tankers in the last decade have drawn security, survey and certification of such equipment in the limelight. But most of these procedures require a high effort in terms of time and personnel. Robots will be a solution to carry out these procedures more efficiently. The use of underwater robots in this field will furthermore open up new possibilities to survey and certify ships or oil production facilities on sea without the need of a dry-dock (which is highly relevant for large offshore production facilities that need to last years on sea). This paper will present some applications of robotic systems that offer a solution for the survey of ship's hulls and the control of the state of large anchor chains. The application of such systems can lead to new possibilities of survey for certification assuring the security of these facilities and ships on sea.

1. Introduction:

Maritime disasters in the last decade have put once more in evidence the importance of periodic inspection and the quality of such assessment. The certification of vessels and large offshore facilities like FPSOs (Floating, Production, Storage & Offloading) are nowadays done mostly by human inspectors. But this task becomes more difficult for the inspection for the underwater sections or, in the case of double hull tankers, the ballast tanks. While in some cases divers and Remote Operated Vehicles (ROV) can be used, in other cases an inspection is simply impossible.

This paper will present three recent projects that address at the issue of inspection, survey and certification of such equipment. Nowadays mainly two means of survey are used: the Non-Destructive Testing (NDT) with for example ultrasonic sensors, to measure the thickness of steel plates, and the visual inspection to detect cracks and evaluate the state of the coating (Fig.1). The three robotic systems presented in this paper bear the possibility to undertake such measurements and evaluation for the certification of offshore facilities.

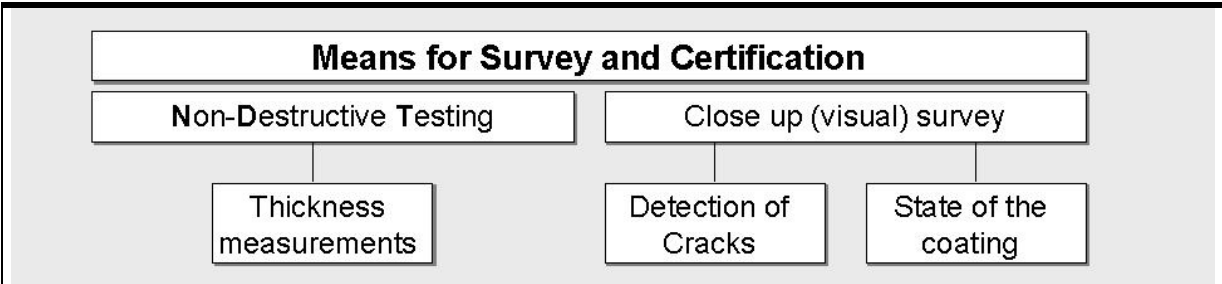


Fig.1: Means of survey and certification

2. Robots for Survey and Certification

Robots can be used for the survey and certification of large offshore structures or ships. While some of these systems can replace the human diver for the inspection work, some others even open up new possibilities of survey. The following sections will present three examples of robots in this field: The OCTOPUS hull crawler for the inspection of the outer hull of these facilities, the ICARE chain climbing robot for the inspection of anchor lines and the ROTIS Remote Operated Vehicle (ROV) for the inspection of the inner parts of double hull tankers.

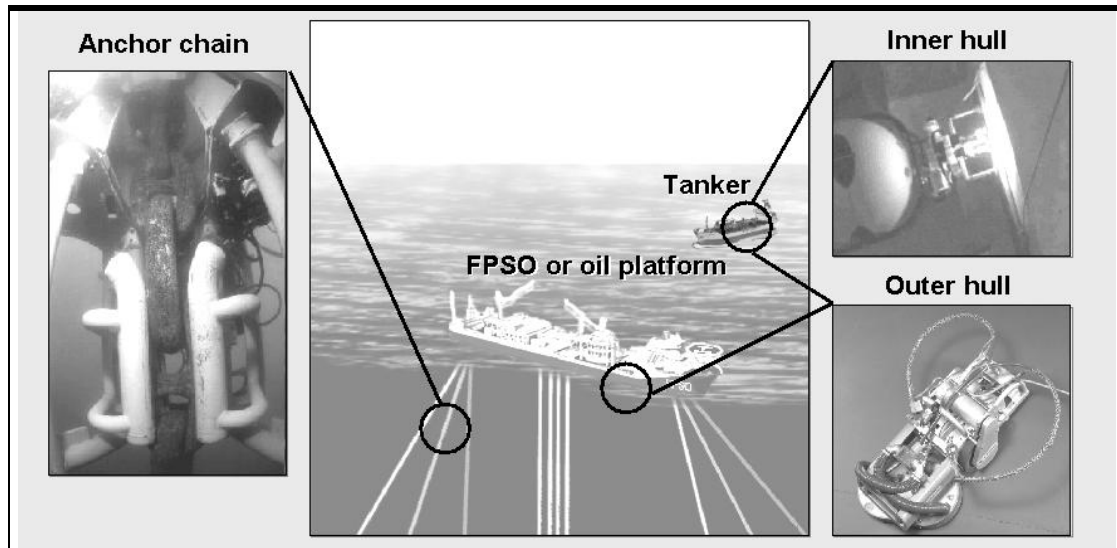


Fig.2: Possibilities for robotic inspection on offshore structures and ships. (left: The ICARE chain inspection robot; top right: the ROTIS inspection ROV, photo courtesy JRC; bottom right: the OCTOPUS hull crawler)

2.1. The OCTOPUS hull-crawler for inspection and thickness measurements

One robot that addresses to the issue of the inspection of the outer hull of ships and other structures like FPSOs is the OCTOPUS. The OCTOPUS is an automatic hull crawler equipped with permanent magnets enabling this vehicle to crawl along vertical surfaces. The carrier vehicle was at the origin developed for the cleaning of ships hulls by High Pressure (HP) water blasting in the dry dock. The consortium working on this project was I.CO.S.R.L (I), R.G.I Resource Group Integrator (I), LISNAVE Shiprepair (P), UNINOVA (P) and CYBERNETIX (F), *Weiss et al. (2003)*.

Several evolutions of this robot were build since this first approach like the OCTOPUS for painting (also the PASOC project) and an underwater version of the OCTOPUS crawler. Especially the last version is of high interest for the subject of inspection of ships or offshore facilities.

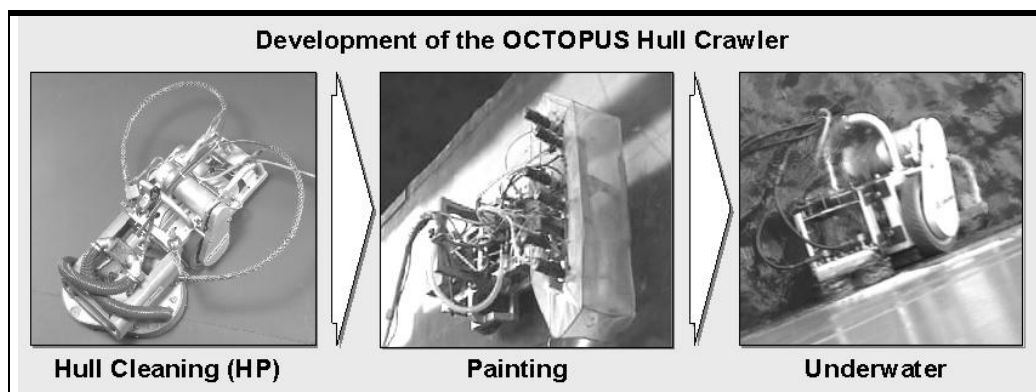


Fig.3: The different applications of the OCTOPUS crawler

In its original version the OCTOPUS was dedicated for the washing and blasting of large surfaces like the ship's hull, *Weiss et al (2003)*. The surfaces that can be treated can be either vertical, inclined or horizontal. The capabilities of the vehicle allow a treatment of approximately 80% of the complete surface of a ships hull (problems arise with small angles like for example at the bow and the stern parts). For the washing of the surfaces an efficiency of approximately 150 m²/h was reached and for the blasting 90 m²/h with a linear speed of 0.15 m/s. For movements without blasting, OCTOPUS reaches speeds up to 0.30 m/s. The blasting is executed with Ultra High Pressure water jetting that

reaches a pressure up to 2500bar. One of the advantages of the system is the fact that the used waters and thus the waste, are recovered by the system. This makes OCTOPUS a very environment friendly tool compared to the techniques used nowadays that imply pollution and overspray. The same can be stated for the noise pollution: Traditional hand-carried water guns emit noise levels up to 115 dB, OCTOPUS came in at 65 dB which also represents a significant ergonomic and ecological aspect for the workers and the environment.

Since the official end of the OCTOPUS project, further applications for the crawler were developed. The idea was always to keep the robot as modular as possible. One crawler can be equipped with different tools and can thus be used for different applications. While the first OCTOPUS was only for the use in the dry dock, a underwater version was developed since then. Fig.4 shows different underwater applications of the robot.

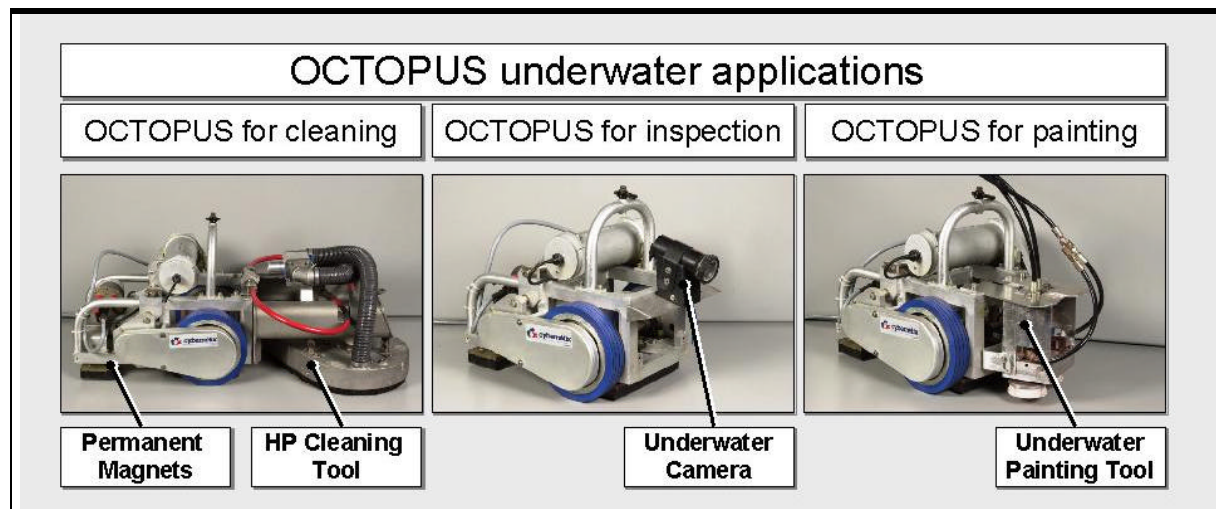


Fig.4: The modular OCTOPUS crawler with its underwater tools.

For the inspection and certification of ships and offshore facilities, the camera version is of high interest. It can be necessary to first clean the surface before doing the inspection. The HP Blasting tool can here be replaced by a system of brushes for underwater applications. One of the main arguments for this is the fact that the equipment for the HP blasting is too large and inflexible for this application. Furthermore the force of blasting is more adapted to the removal of the coating, which is not needed for the inspection. Electrically driven brushes on the other side need a relatively small cable for the power supply and are in this application sufficient, since only marine growth is to be removed, *Chardard (2003)*.

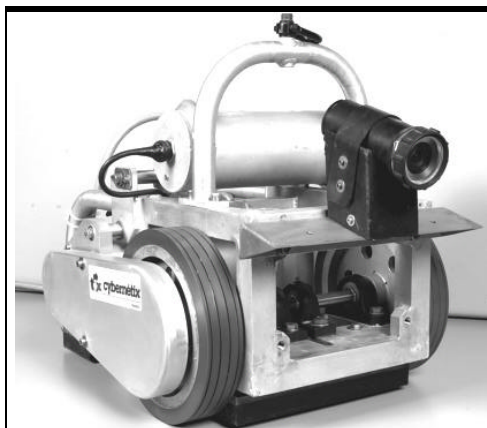


Fig.5: OCTOPUS for underwater Inspection equipped with camera

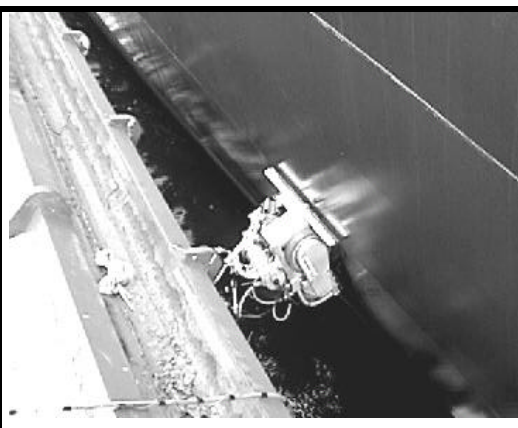


Fig.6: OCTOPUS for underwater inspection during trials on a cruise ship

Experiments with the underwater camera system were executed on a cruise ship in Marseille harbour. While the results were promising, one critical point is clearly the visibility in the water which limits the vision field to the area in front of the vehicle. (This is worsened by the fact that waters in harbours are normally not very clean, so the system is confronted with the worst situation imaginable.) The positioning of the vehicle is a further challenge for the execution of a hulls survey. The steel plates where damages were found must be located correctly in order to foresee replacement or further inspections. Therefore the inspector must exactly know where the vehicle is at the moment of the survey.

The procedures for the inspection of a ships structural and equipment requirements foresee that it may be necessary to check the underwater portions of a ship in order to evaluate its safety, *Paris MOU (2000)*. A typical intervention scenario for such an inspection by a underwater OCTOPUS is shown in Fig.7: The robots crawls along the sub-sea hull of the tanker while the inspection crew is following the robot by a small support vessel to avoid long cable lines.

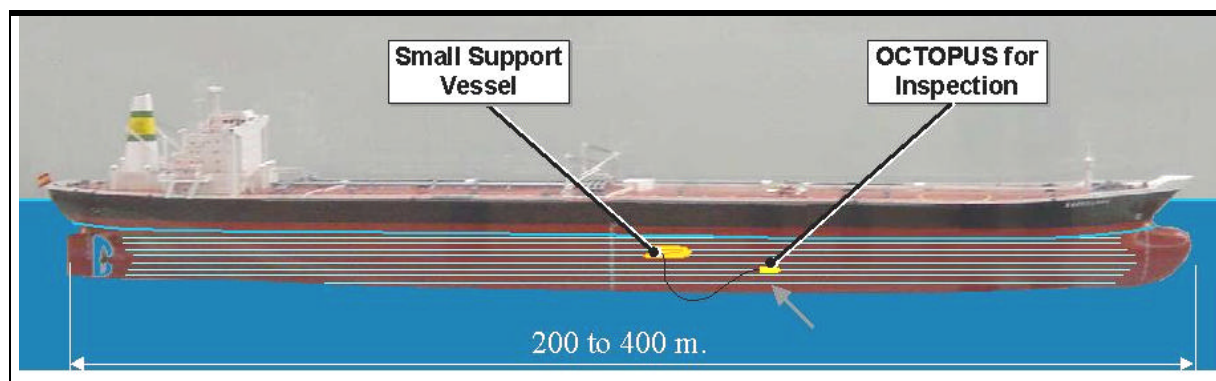


Fig.7: Inspection of the outer hull of a tanker.

Thickness measurements of the hulls walls are a essential part of such inspections, since this data informs about the seaworthiness of the ship. Such data is in most cases acquired through non-destructive testing (NDT) with ultrasonic sensors. One of the future developments in the frame of the OCTOPUS will be to equip the robot with NDT sensors in order to enable it to take thickness measurements along the hull.

This proposed procedure, or the possibility to execute inspection while the ship is still in water, is of high interest in the case of oil platforms or FPSOs, thus structures that cannot easily be placed in a dry-dock and that need to last for years on sea.

A scenario imaginable would be, for example, to do the inspection while the ship is on its way to the dry-dock and thus preparing a planning before. This possibility would avoid long standing times in the dock, since the reparations and the necessary material can be foreseen before the arrival of the ship.

2.2. The ICARE chain climbing robot for the inspection of anchor lines

A particular application of automated inspection systems can be found on FPSOs (Floating, Production, Storage & Offloading; Permanently installed oil production vessel): These floating production facilities are fixed with mooring chain lines to the seabed. To our knowledge there does not exist any automated system so far that can be used to measure these chains, while its integrity is of high importance for the security of these facilities. Typical testing procedures include close visual inspection of the chains, enhanced representative NDT sampling and dimension checks, *IACS (1995)*. In the past, incidents happened due to broken anchor lines, leading to a drift off of stockade buoys.

The ICARE system is a remote operated system developed by CYBERNETIX for surface and sub-sea cleaning and inspection of anchor chains. The robot takes pictures of each chain member that allow

the inspector to check the chain for cracks and corrosion. A inspection chamber that comprises the vision system turns around each chain member in order to take frontal pictures of each member (since those are shifted by 90° each). The certification will be supported by an image processing system that analyses 12 dimensions on each chain member (see Fig.9). The so acquired data is presented on the Man-Machine-Interface to the inspector, enabling to establish a detailed report about the state of the chain. Additional measurements can either be pre-programmed or manually chosen during the process. Onshore inspection results showed an average error of 0.8% (less than 1mm) for the measurements in the case of chain sizes up to 6".

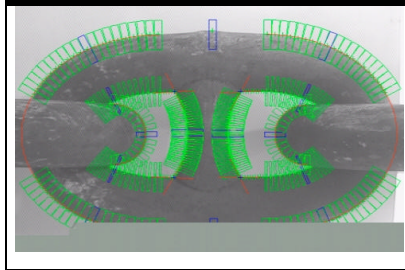


Fig.8: Still image of a chain member taken by ICARE

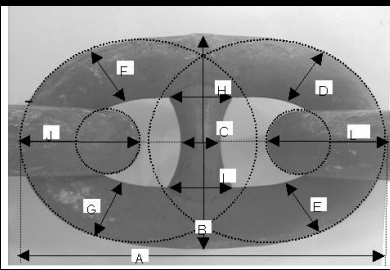


Fig.9: 12 measurements are take for each chain member

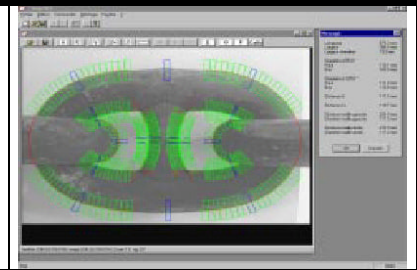


Fig.10: The Man-Machine-Interface of ICARE

The vertical displacement of the system, or the "climbing" up and down the chain, is done by a system of two claws: One claw holds the chain while the second one is displaced. When the second one closes around the chain, the first one can be displaced. The whole system is slightly buoyant in water. ICARE can be operated in stand-alone configuration or be interfaced with a Work ROV that supplies the system with hydraulic and electric power and assures the data transfer to the surface.

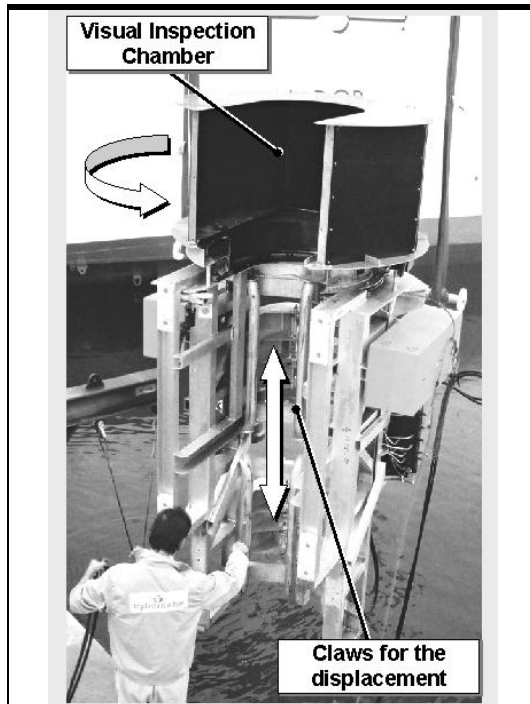


Fig.11: ICARE being lowered into the water



Fig.12: Underwater photograph of the ICARE climbing down the anchor line.

Like for the inspection of the outer ship hull with the OCTOPUS robot, cleaning of the anchor chain could be necessary in order to take correct measurements. A high pressure water jet system was interfaced to the ICARE for this reason, enabling to clean the chain members before the inspection.



Fig.13. and 14: Cleaning of the chain members by HP water jet

The initial prototype that was developed in 1999 was successfully tested at the CYBERNETIX facilities and lead, due to a positive feedback of potential industrial end users, to an improved industrial version of the ICARE anchor chain inspection system.

2.3. The ROTIS robot for the inspection of double hull tankers

Following to the tragedy of the PRESITGE accident in November 2002, single hull tankers were banned from the EU waters, *Andritsos-Maddalena (2003)*. The introduction of double hull vessels is widely seen as appropriate mean to avoid future maritime catastrophes of this kind. However new means to inspect and monitor those vehicles must be found in order to assure maritime safety efficiently.

In the frame of the ROTIS project, a **R**emote **O**perated **T**anker **I**nspection **S**ystem was developed, based on a compact, free-floating ROV able to navigate inside the flooded ballast tanks of double hull tankers to carry out close-up visual inspection and wall thickness measurements, *Andritsos-Maddalena (2003)*. The consortium developing the ROTIS was co-ordinated by TECNOMARE (I) with ZENON (GR), ENEA (I), LLOYDS (GR), CS&A (GR), AVIN (GR), HUT (FIN) and JRC (EC) as project partners. The project was financed by the European Commission under the BRITE-EURAM framework.

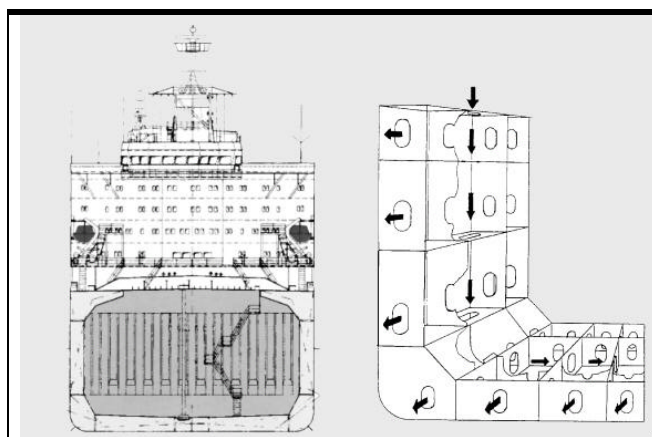


Fig.15: Ballast tanks give access to virtually all the structural parts of a double hull vessel



Fig.16: Photographs from the ballast spaces of a new build double hull tanker

Fig.15 shows the basic concept of the ROTIS system: A highly manoeuvrable, small ROV navigates through the manholes inside the ballast tanks of the ship. It is linked through a tether to an intermediate unit situated on the ship's deck, which is radio linked to a surface unit in the ship's bridge.

The ROV is equipped with a ultrasound measuring probe and a vision system allowing the operator visual inspection and thickness measurements of the vessel's walls and stiffeners. Following figures show 3D models of the ROTIS while navigating in the ballast tanks and photographs of the ROV taken in the JRC test pool during the trials. A specially designed mock-up of a double hull tanker section allowed intensive tests of the equipment at the JRC. These test confirmed the global concept of the robot especially in terms of navigation and manoeuvrability. The thickness measurements were identified as a critical aspect since it requires a stable, perpendicular docking to the walls. Suction pads were used to fix the ROV to the wall. To improve the measurement a wire-brush was installed in order to clean the surface.

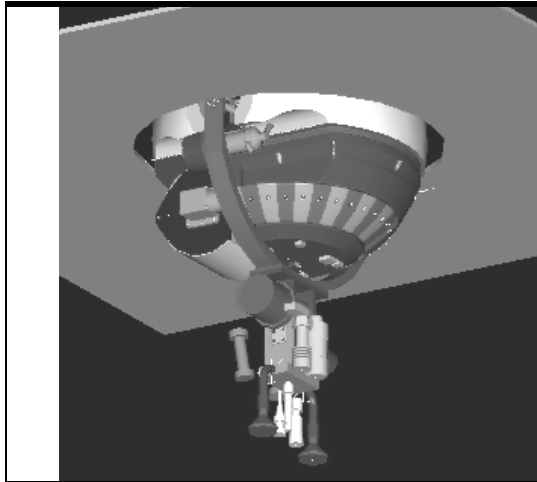


Fig.17: ROTIS passage through a 60 cm Ø manhole

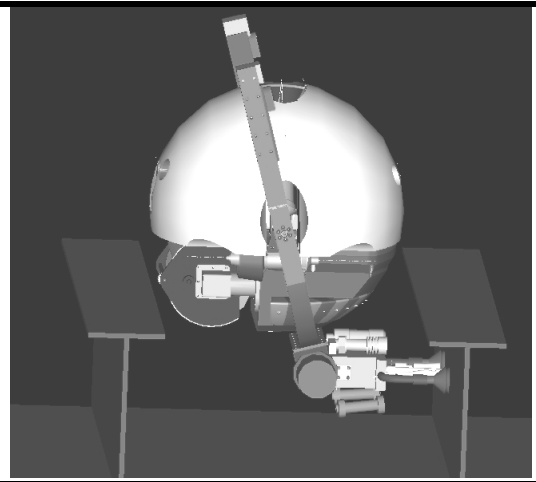


Fig.18: 3D model of ROTIS inspecting a stiffener



Fig.19: Photograph of ROTIS passing through a horizontal manhole



Fig.20: Photograph of the docked ROTIS while perform visual inspection and wall thickness measurements.

The ROTIS project confirmed the feasibility of remote operated inspection of double-hull vessels which presents the possible answer to a economic certification of those ships for the future.

3. Conclusion

The paper presented three robotic systems that could deliver an answer to the need of automated inspection of offshore facilities and ships. While these procedures are nowadays mainly executed by humans or divers, robots could be used in the future for these tasks. The advantages resulting from this are various: Robots can replace divers in this difficult work, leading to more precise and standardized measurement methods and even new possibilities to certify the security of these facilities.

Acknowledgements

The authors want to thank the members of the ROTIS consortium for their helpful input for writing this paper. ROTIS was a project financed by the European Commission under the BRITE-EURAM framework , coordinated by TECNOMARE (I) with ZENON (GR), ENEA (I), LLOYDS (GR), CS&A (GR), AVIN (GR), HUT (FIN) and the JRC (EC) as project partners.

The OCTOPUS robot was developed under the GROWTH programme of the European Commission. The project was coordinated by CYBERNETIX (F) with I.CO.S.R.L (I), R.G.I Resource Group Integrator (I), LISNAVE Shiprepair (P) and UNINOVA (P) as project partners.

References

WEISS, P.; SCHOM F.; CHARDARD Y. (2003), *Intervention of robots in shipbuilding and maintenance*, 2nd Int. EuroConference Computer and IT Applications in the Maritime Industries, Hamburg, pp.425-431

CHARDARD, Y. (2003), *Robotics in ship maintenance and inspection*, Offshore/Marine Fabrication & Conversion 2003, Singapore

Paris Memorandum of Understanding on Port State Control (2002) , adopted 9th May 2002

International Association of Classification Societies IACS (1995), Guidelines for the Survey of Offshore Mooring Chain cable in Use (No.38), 1995

ANDRITSOS, F.; MADDALENA, D. (2003), *ROTIS: Remotely operated tanker inspection system*, 8th International Marine Design Conference, Athens

Research on Industrial Engineering System by using Wearable PC

Yuichi Sasaki, Mitsubishi Heavy Industries Ltd., Nagasaki/Japan, yuichi_sasaki@mhi.co.jp

Hiroyuki Yamato, University of Tokyo, Tokyo/Japan

Shoichi Enomoto, University of Tokyo, Tokyo/Japan

Trial applications of a 3D simulation tool for the shipyard showed that simulations with a virtual human model is useful for work improvement. The development of Wearable PCs allow now highly accurate work measurement in the shipyard. Then, the Industrial Engineering system for the shipyard will be achieved by using this technology. Then, in this research, the system concept of Industrial Engineering system for the shipyard, and the prototype of the system were stated.

1. Introduction

Industrial Engineering is the activity to design, reform and define the integrated process in the consideration of man, material flow and facilities:

- 1: Measurement of the work in the factory
- 2: Analysis of the result of measurement
- 3: Improvement of the activity in the factory.

Industrial Engineering is used as a tool for work improvement of the production systems for easier, faster, and more economic work. In mass production industries, industrial engineers measure workers and facility operation and study work improvement in assembly simulations as shown in Fig.1.

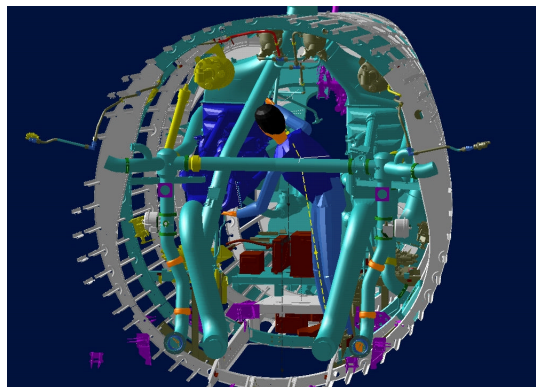


Fig.1: Work improvement in mass production industries

In shipbuilding, the same general principle can be applied:

- 1: Measurement of the work in the shipyard
- 2: Analysis of the result of measurement
- 3: Improvement of the activity in the shipyard

Figs.2 and 3 show the trial result of Industrial Engineering in shipyards. The sequential operation measurement, Fig.2, shows the continuous motion of the facility and worker. The work sampling method, Fig.3, is the analysis of the facility and human motion at regular intervals. In this way, the actual work of the shipyard can be measured and analyzed. Two problems made application of Industrial Engineering methods difficult for shipyards:

1. Facilitation of the work measurement

The work measurement of 1 cycle takes very long time in shipbuilding. Also, it is difficult to measure all the work by a few people, because it is very complicated. Then, it seemed to cost very much to measure the work in the shipyard.

2. Quantitative evaluation of work improvement

It is difficult to evaluate the work improvement quantitatively, because shipbuilding is production on order.

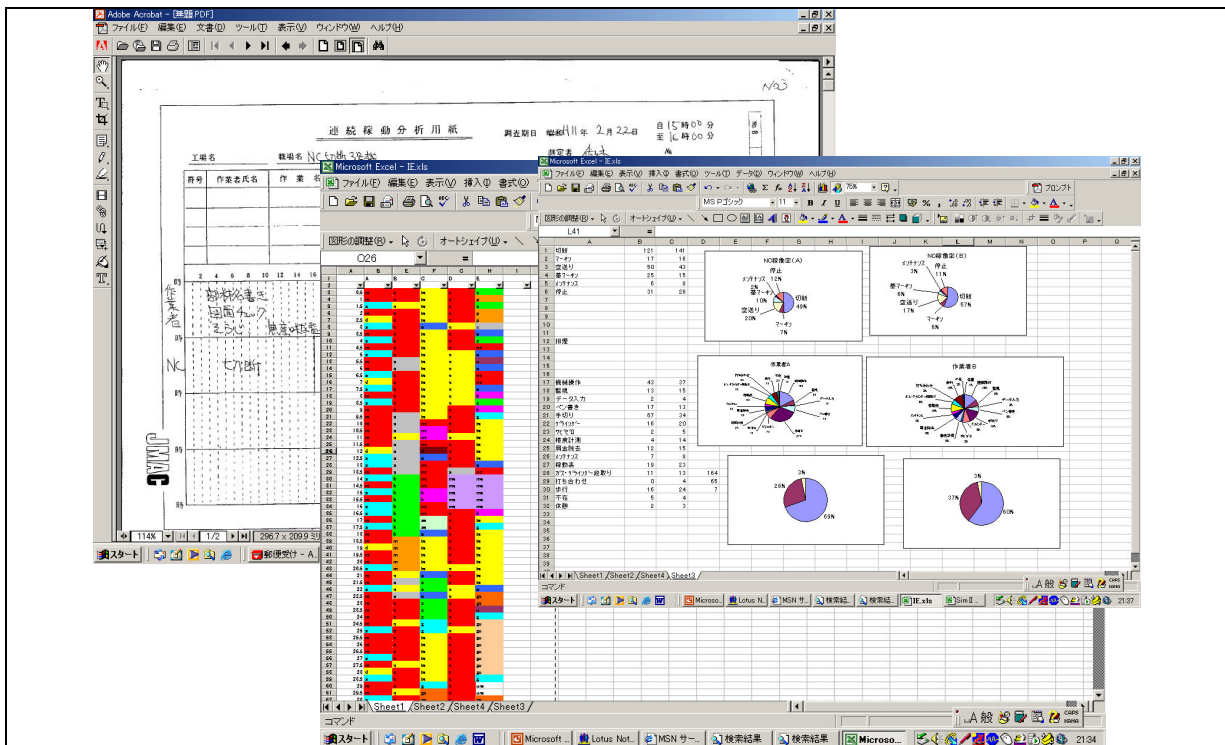


Fig.2: Result of Industrial Engineering for Shipyards (sequential operation measurement)

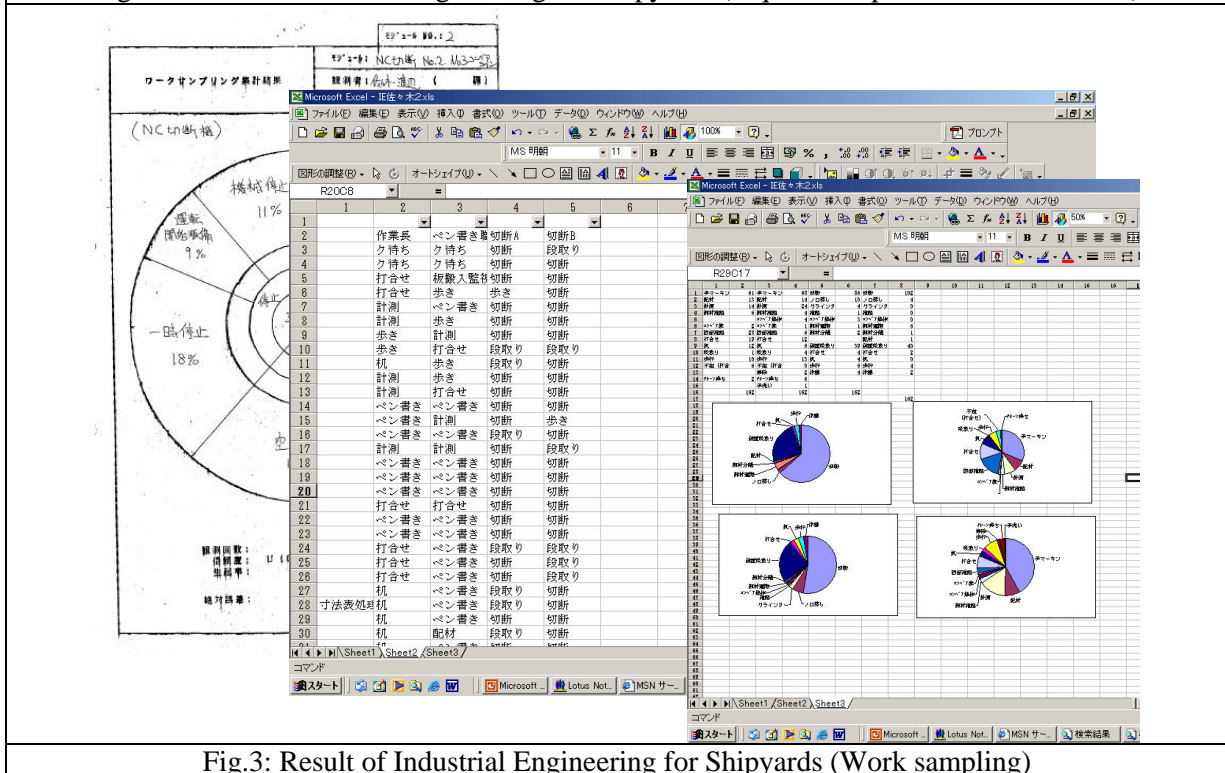


Fig.3: Result of Industrial Engineering for Shipyards (Work sampling)

Wearable PCs, Fig.4, were developed recently. Fig.5 shows the application of assembly simulation system to the shipyard. This system enables to evaluate the workability in the computer. It can evaluate the work improvement quantitatively by compare the multiple work procedure, *Sasaki et al. (2001), IMS (2001, 2002)*. These technologies allow to solve the above stated two problems:

1. Facilitation of the work measurement

It is easy to measure the complicated work by using multiple wearable PC in the network.

2. Quantitative evaluation of work improvement

It will become possible to evaluate the work quantitatively by using assembly simulation systems.



Fig.4: Wearable PC

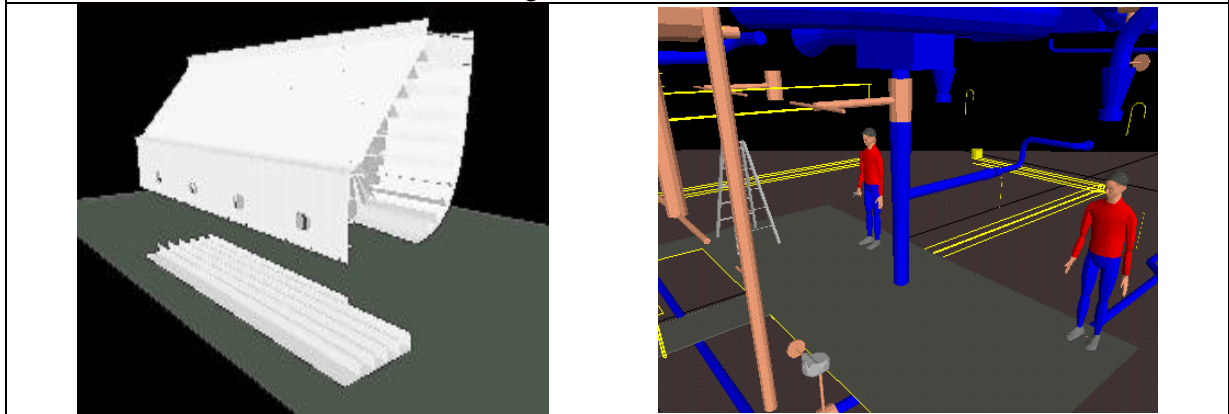


Fig.5: Assembly simulation systems for shipyard

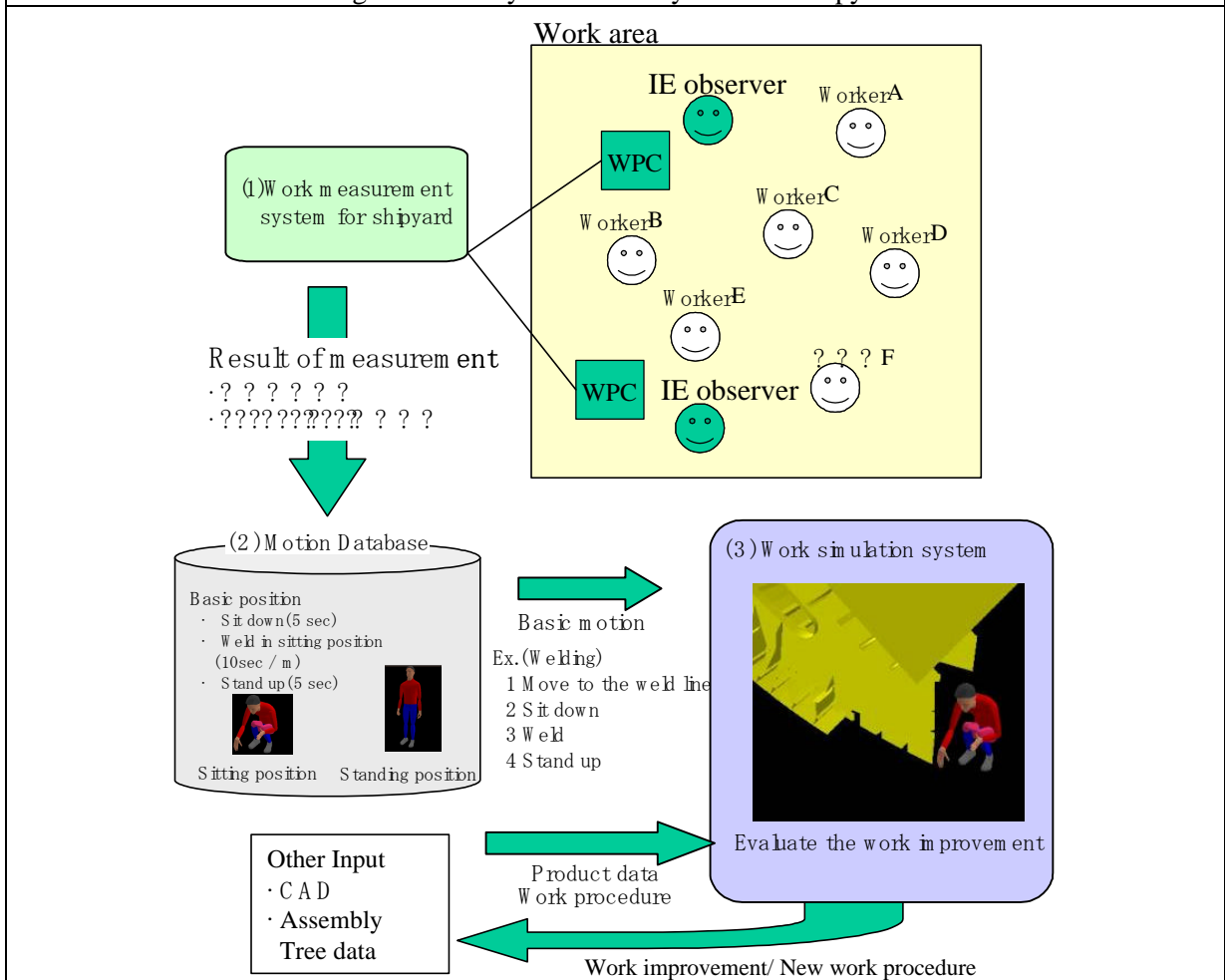


Fig.6: Concept of Industrial engineering system for shipyard

2. Industrial engineering system for shipyard

Fig.6 shows the system concept of Industrial engineering system for shipyard. This system consists of 3 sub-systems:

- (1) Work measurement system by using Wearable PC
This sub-system measures the work in the shipyard by using Wearable PC. It has two functions to measure the work. One is the sequential operation measurement function and the other is the work sampling measurement function. The result of the measurement is analysed and stored in the motion database.
- (2) Motion database
This is the database of work motion. The data consists of work item, work time and workers motion. This can be defined by using Japan Management Association Management Center (MOST) approach instead of the result of work measurement.
- (3) Work simulation system
This system can simulate the workability easily by using motion database and actual product data. Then we can evaluate the work improvement by comparison of many cases.

This concept enables the quantitative evaluation of work improvement in the shipyard by using actual performance in the work place. We will thus realize an industrial engineering system for a shipyard. In this research, '(1) Work measurement system by using Wearable PC' and '(3) Work simulation system' were studied and a prototype system developed.

3. Prototype system

3.1. Work measurement system based on Wearable PC

This system, Fig.7, measures the workers motion in the work site. In this system, work item is shown in the Wearable PC, and the workers motion can be recorded by choosing this work item. Two functions are developed in this system:

- (1) Sequential operation measurement function
This function records the work item and start time of the work activity to measure the sequential work in the factory. The end time of the work activity is automatically defined by recording the start time of next activity. Fig.8 shows the result of measurement. In this way, sequential operation can be recorded by using this function.
- (2) Work sampling measurement
This function records the work item at regular intervals (e.g. 30 seconds). It can measure many objects (workers and facilities) simultaneously. An example of multiple measurement is:

Worker A: every 30 seconds	0sec, 30sec, 60sec, 90sec,...
Worker B: every 30 seconds + 10seconds	10sec, 40sec, 70sec, 100sec,...
Facility C: every 30 seconds + 20 seconds	20sec, 50sec, 80sec, 110sec,...

This function can analyze the percentage of work item allowing to evaluate the percentage of extra or unproductive work.

3.2. Work simulation system

Fig.9 shows the system image of the work simulation system. A motion database is used as the basic motion in the virtual shipyard. Output of the computer aided process planning system, *Sasaki et al. (2002)*, generates required product data. These data are used to define the work procedure allowing us work simulation. By using this system, work time of each work procedure can be calculated and we can evaluate the work improvement quantitatively.

入力してください 残り17秒 経過時間[02:23] 現在時刻[09:27:30]

対象	場所	動作		
対象C	作業台A	機械操作	屑金除去	休憩
	作業台B	監視	メンテナンス	動作名14
	作業台C	ペン書き	図面	動作名15
	ET	手切り	打ち合わせ	その他1
	NC	グラインダー	歩行	その他2
	クレーン	精度計測	不在	その他3

履歴

直前	前々回	3回前
対象B, NC, メンテナンス, 時刻09:27:17	対象A, 作業台B, 打ち合わせ, 時刻09:27:11	対象C, クレーン, 機械操作, 時刻09:26:47

決定

停止

×

LA 般 風 風 CRPS 10000

Fig.7: Work measurement system

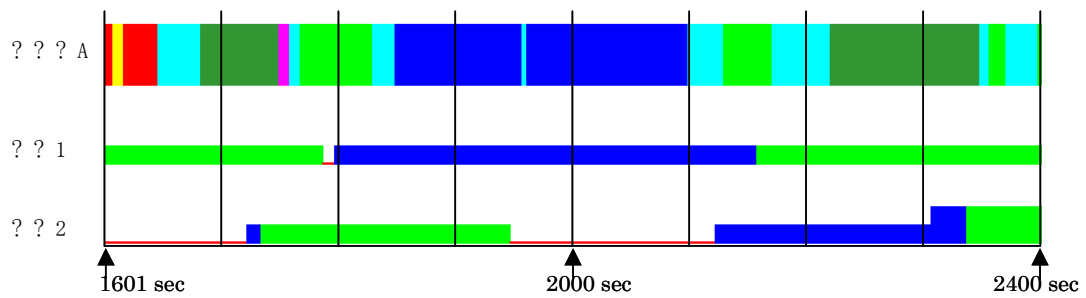


Fig.8: Example of measurement

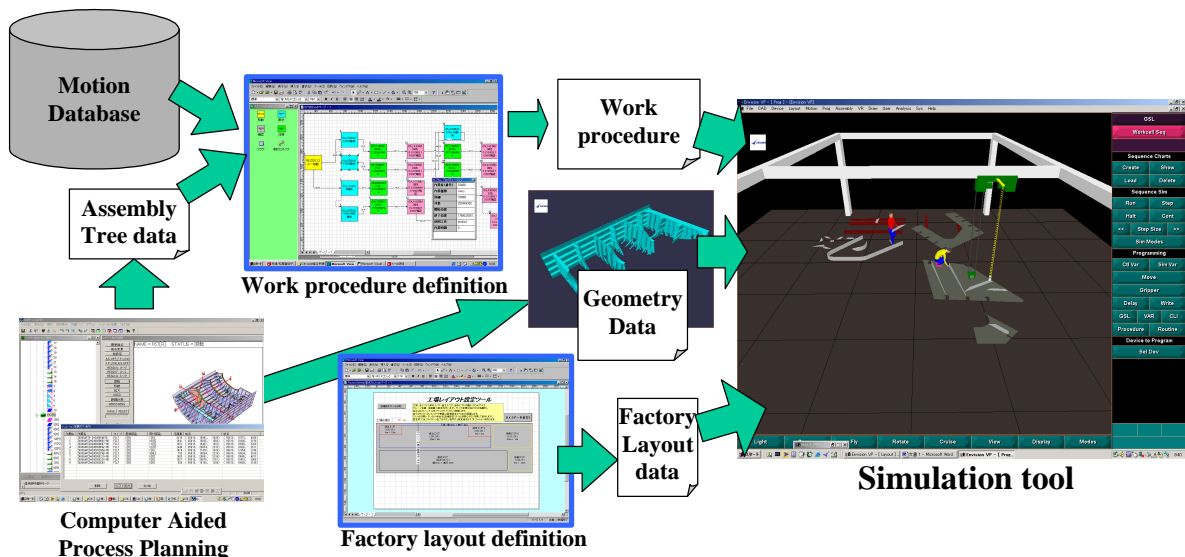


Fig.9: System image of work simulation system

4. Next steps

In the next phase, the development of motion database is needed and then, industrial engineering system for shipyard will be realized. In the future, automation of the work measurement, standard work time calculation function and scheduling function by using the result of IE measurement will be studied.

5. Summary

An industrial engineering system for shipyard was studied to evaluate the work improvement quantitatively. A concept based on three 3 sub-systems was proposed:

- (1) Work measurement system by using Wearable PC
- (2) Motion database
- (3) Work simulation system

Prototype systems for (1) and (3) were developed and their usefulness verified.

Acknowledgements

The authors would like to express their gratitude to IMS IRMA project for the development of work improvement system in the IRMA project. And authors also would like to express their gratitude to Ship and Ocean foundation, Japan Marine Science Inc. and Dr. Hideyuki Ando for the development of work measurement system by using Wearable PC.

References

- SASAKI, Y., MIURA, M., TAKANO, G., FUJITA, K., FUJIWARA, N., IIDA, A., SAGOU, A. (2001), *Research on total cost evaluation system for shipyard*, 7th Int. Symp. Japan Welding Society
- SASAKI, Y.; SONDA, M.; ITO, K. (2002), *A study on 3-D digital mockup systems for work strategy planning*, ICCAS'02, Malmö
- SASAKI, Y., SONDA, M., ITO, K (2002), *Development of computer aided process planning system based on knowledge base*, J. Marine Science and Technology

Mobile and Web-based Services for Ship Operation and Survey: A Portal-centric Approach

Patrick Hupe¹, Uwe Langbecker², Mubashir Aziz¹, Joachim W. Schmidt¹,

¹Software Systems Institute, Technical University Hamburg-Harburg, Hamburg/Germany,
{pa.hupe, mubashir.aziz, j.w.schmidt}@tuhh.de

²Germanischer Lloyd, Hamburg/Germany, lgb@gl-group.com

Abstract

It is highly desirable that ships operate safely 24 hours a day, 7 days a week. To achieve this, ship classification societies supply ship owners and operators with suites of IT-supported services, increasingly web-based and highly personalized. Access to such services is required worldwide and timely by ship owners, ship surveyors and head-office staff. Depending on connectivity and context, a variety of different working environments has to be supported: offline and online, mobile and stationary, public and private etc.

The project reported here investigates suites of services for several classes of use cases:

- *web-based ship information and calculation services for classification society staff,*
- *web- and mobile interface-based ship status checking for ship owners, and*
- *offline and re-synchronized task execution for ship surveyors with mobile devices.*

State-of-the-art web and mobile system technology has proven to be suitable for the design and realization of integrated portal systems supporting such suites of services. The results presented here are the outcome of a joint national R&D project.

1. Introduction

Maritime industries, as most globally acting service providers, apply modern information and communication technology to streamline and optimize their essential business processes. Such process enhancements require immediate access to information and services via globally available company access points. Enterprise Information Portals (EIP) realize such global access points by exploiting off-the-shelves Web technology.

Enhancement of work processes in the maritime industries is, however, often restricted by two kinds of limitations of current EIP solutions:

1. EIP technology provides access mainly for desktop users in a *stationary* environment, i.e., for company staff or customers in their offices equipped with Web browsers and standard office software. Most EIPs do not provide, however, appropriate interfaces for users in *mobile* work environments. In addition, the very nature of Web-based portals necessitates portal users to maintain continuous online connection, and there is no sufficient work support for combined online/offline use. However, work processes in ship newbuilding and ship survey inherently require mobile solutions.
2. While portals focus on access to large amounts of electronic artifacts, they usually lack continuous support for long-term, intra- and inter-organizational processes. There is, however, an increasing need to support communicating user groups in stationary as well as in mobile environments. Such synchronized work support is required on a variable time granularity scale ranging from instantaneous transactions to complete order processing from order initiation to order completion and payment. Process artifacts need to be archived for subsequent inspection of process flow and history.

Example from the maritime industries include *classification societies* which handle and control a large variety of business processes such as customer requested information services or collaborations in ship newbuilding and ship surveying. A ship survey process, for example, spans multiple

organizations and organizational units including ship owners, society head offices as well as ship surveyors working in virtually any harbor worldwide.

Users in different roles require personalized access to information and services customized to their specific work needs. The following scenarios give a first idea of online and offline, mobile and stationary work environments:

- Customers want to review status information about their fleet, issue requests for ship survey and contact surveyors at ports either from their offices or while traveling.
- Classification society staff members use ship operation-related services at their office but may also request information in mobile environments from PDAs, mobile phones, etc. For example, a special “Emergency Response Service” which integrates access to complete information about a ship including construction details is of particular importance in extreme operational situations. Today such services are already available at Germanischer Lloyd.
- Surveyors perform planned periodical surveys as well as unscheduled damage surveys on short notice. Such service processes are undertaken world-wide onboard ships and require intensive support of the surveyors’ mobility.

In this paper, we present portal-centric and mobile solutions for ship operation and survey which have been analyzed, designed, realized and tested in a joint national research project conducted by the Software Systems Institute of Technical University Hamburg-Harburg and Germanischer Lloyd, *Hupe and Schmidt (2004); Langbecker (2004)*.

The paper is structured as follows: In section 2 the research project’s main goal, the provision of an online service portal system and three domains of research are described. In section 3, support for mobile online and offline service is outlined. Section 4 presents future work and section 5 closes with concluding remarks.

2. An Online Portal for Selected Ship Technology Services

The research project “Internet Services for Ship Technology” focuses on realizing an information and service portal for selected application areas for Germanischer Lloyd staff, surveyors and customers.

The service and information portal is built on top of a portal platform, which connects two views on information: From the users’, or *information consumers*’ point of view, it provides customized and personalized access to information, while from the information sources’ perspective, in this case *information repositories* like databases, XML document repositories, document management and content management systems, it integrates information sources and allows interrelating information assets of different provenance, Fig.1.

The portal platform, *Wegner (2002)*, *infoAsset (2001)* provides an infrastructure for the realization of portal-based information services and is used in several research projects, e.g. *EURIFT (2003)*. It is based on Java technology that makes it suitable for deploying portal services on several standard hardware and operating system configurations.

The platform provides authentication and authorization services and facilitates the realization of personalized and role-based services. It can integrate various information repositories by abstracting from the concrete repository type and vendor, which allows integration of a multitude of information sources. Information sources can be relational databases, document management systems or XML-repositories.

Application domains

Ship newbuilding classification and *ship operation and fleet support* are key domains for business activities of a classification society. These domains were selected as application domains in the

research project. Three major research areas have been targeted to realize a portal-centric service infrastructure which can improve existing services as well as provide new and innovative services. The research areas will be described in the following subsections.

2.1 Information management

Companies gather and manage business relevant information in different information systems: documents, e.g. reports, certificates, are typically stored in file systems or document management systems (DMS), while entity data, e.g. ship information and additional items like due dates, schedules and deadlines are usually stored in relational database management systems (RDBMS). Additionally, content management systems (CMS) are used to manage multimedia content such as images, video, audio files, news items and to publish them to a web site or to issue notifications, e.g. newsletters.

A corporate information portal provides a single point of access to enterprise information. It must provide means to search and find information and explore its context, i.e. semantically near information. To make information accessible, it must be retrievable, e.g. via a search mechanism, and it must be traversable. A user must be able to navigate within its semantic context to explore related information structures. Furthermore, information must be extensible, i.e. a user must be allowed to interrelate and annotate given information.

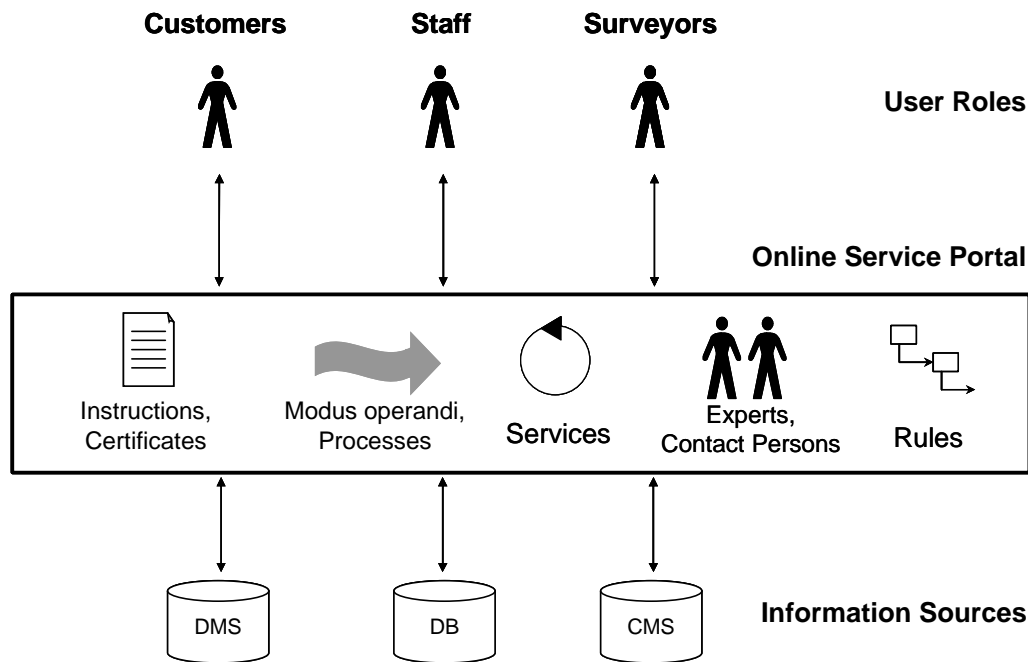


Fig.1: Online Service Portal

The project focused on management of ontological concept and information structures, Fig.2. Users can explore, navigate and annotate the knowledge base, find related information including documents and contact persons. So-called *form sheets* comprising of ship survey instructions and detailed survey items used by surveyors are taken as an example. Survey items can be retrieved via their identification number (“SI 529”), and descriptive text (“oil heating”). Additionally, a user can navigate according to different classifications, e.g. the form sheet structure based on identification numbers and survey area classifications. He can annotate forms and survey items to provide additional documentation and explanation. Apart from this, persons and organizational units (e.g. a department responsible for propulsion systems) can be assigned as contact or expert for a given survey item and form.

As a result, users can easily find information using different search criteria, and they can navigate the enterprise-specific taxonomic and concept structure. Having accessed an artifact, they can browse the

artifact's context and extend the context by linking additional artifacts. Finally, they can request to be notified of changes.

The screenshot shows the WIPS interface with a navigation menu on the left and a main content area. The main content area displays a hierarchical structure of forms and survey items. The tree view shows a hierarchy starting with F101, which branches into F101.3, F101.4, F101.5, and F101.6. F101.3 further branches into F101.3.3.3, F101.3.3.8, SI 290, SI 292, SI 295, SI 1977, SI 1978, SI 2105, SI 2106, and SI 2107. The detailed view of SI 1978 shows its description, explanation, and a list of related survey items.

Concept Classification, Navigation

Concept Description

Concept Extent

Related Persons, Artifacts

Fig.2: Classification, structuring and annotation applied to forms and survey items

2.2 Service Integration

Computers support users at work. Typically, users perform tasks by handling electronic artifacts, i.e. create and edit documents, send mails, make appointments, etc. The technical means to handle artifacts are applications and tools. We generalize applications and tools as user-oriented *services*, Jagoe (2003). Services are a uniform view on means to add value to artifacts.

While many companies use standard applications like word processors and spreadsheet applications, in the maritime industry many industry-specific tasks require tailor-made tools. Take, for instance, a ship newbuilding project: A newbuilding project comprises of many artifacts, e.g. project documents that describe machinery, engine, hull, etc. Services are used to manipulate the artifacts. Such services can be document-specific editors, validators, or visualization tools.

Many organizations and organizational roles are involved in a ship newbuilding project, e.g. ship owner, classification society, shipyard, manufacturer, component suppliers. These different users should be enabled to work with the project documentation, e.g. review, inspect or edit it. Some might only have a limited, read-only view on documents, while others are allowed to change them. Project participants should be notified of changes.

We want to realize a service-oriented work environment that outstretches from the portal to the user's computers. At a first level, we provide a view on artifacts. This is quite simple to realize using standard Web technology. Provision of services is a more complex task, which leads to complex maintenance problems: services must be installed and maintained on the users' computers. Furthermore, as many organizations are involved, the distribution of services cannot be managed by a single company's IT administration. We have overcome these problems by defining and maintaining services centrally on the portal and deploying them locally on-demand to the user, ensuring that service updates are transparently propagated to the users.

Provision of services and artifacts defines an encompassing service-oriented work environment where a user can handle artifacts using the appropriate service. When a user decides to work on an artifact, the service is made available on his computer. The artifact is loaded from the portal and can be edited.

After finishing work, the artifact's new version is published to the portal, thus made available to other users. The portal's authentication and authorization mechanisms ensure that only authorized users are granted access to services and artifacts, Fig.3.

Existing services used for ship newbuilding projects at Germanischer Lloyd have been taken as examples. These include calculation and evaluation services that are used to validate components of the propulsion system against relevant rules for classification and construction, *GL (2004)*.

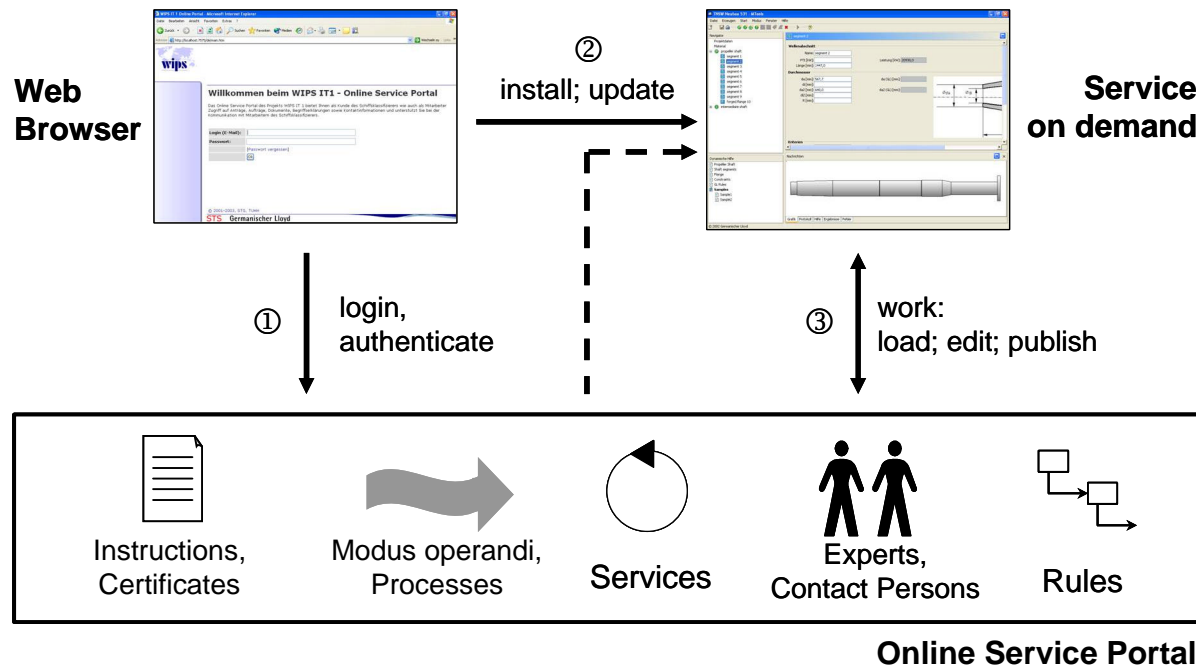


Fig.3: Service Provision on Demand

At a technical level, the provision of a service-oriented work environment based on a portal implies a tight coupling of the constituent parts: service functionality and work artifacts. A work environment is only usable if the following steps are seamlessly integrated: service deployment and invocation; artifact provision, handling and publication and service life-cycle management.

Solutions for distributed installation and maintenance of software exist for different runtime environments and operating systems, *Carzaniga et al. (1998)*: Java WebStart allows web-based installation of applications realized in Java; Microsoft offers the Microsoft Installer (MSI), which permits a simple installation on Windows-based systems. Similar technologies exist for other platforms and operating systems. We have chosen *Java WebStart* for being an operating system neutral standard. It provides a mechanism to transparently update Java software on users' computers. On the client side, Java WebStart only requires a *Java Runtime Environment (JRE)* to be present. Nowadays, the JRE exists for most standard OS platforms (Windows, Solaris, Linux, Mac OS) and is used in many companies.

2.3 Process Support

Support for structured, long-term and distributed processes is a strong requirement in many business domains, *Hupe (2000)*. While enterprise portal systems usually help in artifact creation, editing, annotation and consumption, a lack in comprehensive process support can often be identified. Portals typically focus on serving isolated information requests. If portals offer process support, they typically serve a single user in one or a number of sequential sessions. For instance, requirements for process routing or task delegation are usually not identified.

We take the ship survey process, which is a core process of a classification society, as an example to analyze requirements and to define a suitable representation and execution environment for it, Fig.4.

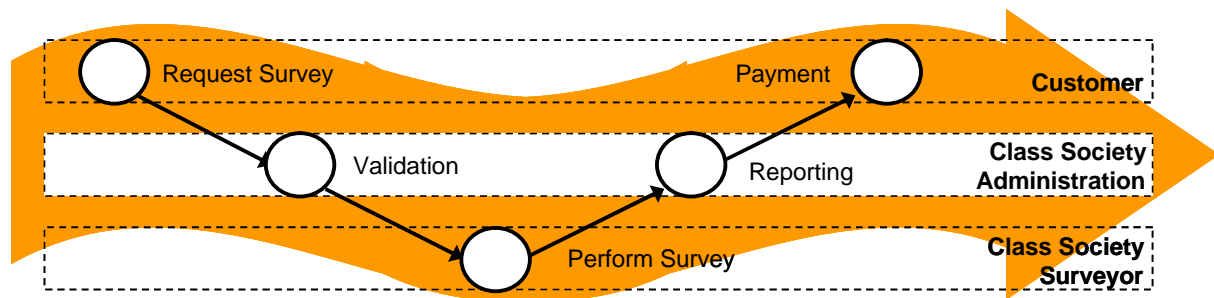


Fig.4: Ship survey process; participating roles

The goal of a ship survey is physical checking of a large number of survey items onboard a ship. Every seagoing ship needs to be surveyed periodically. The ship owner is responsible for requesting surveys. These requests are sent to the head-office of the classification society where they are validated. After validation they are delegated to one or a group of surveyors at a port office in charge. There, the physical inspection is executed and reported to the ship owner and society head-office. The head-office records the inspection results and invoices the ship owner, who makes a payment.

It is crucial to support the distribution of processes over space and time: The ship survey processes are coordinated by society staff from head-office in Hamburg, Germany. Customers (ship owners) monitor the progress in ship newbuildings and surveys which take place all over the world. Surveyors typically work for an area office at a local port. The area offices are distributed worldwide.

A need to support localized work environments can be identified, e.g. the survey items must be provided to surveyors at the area office to enable them to conduct the survey locally and later submit the survey results to the corporate portal. Information synchronization issues must be handled appropriately and according to work requirements. Additional requirements are that one cannot assume the surveyors in an area office to be permanently connected over the Internet with the society's service portal located at the head office. To summarize, every process participant must have access to relevant services and information while, at the same time, offline and online work environments must be supported. The additional aspect of user mobility will be covered in section 3.

The need to provide services and information that are relevant for a person in a local work context and offline work support has led to a solution where *personal information portals* serve as a localized and personalized view on the corporate central information portal with respect to work items, Fig.5. A personal information portal provides the same services as the central portal, while maintaining only artifacts locally that are relevant in the user's work contexts. Work description artifacts and information artifacts are synchronized from the central portal to the personal information portal, *Lehel (2002); Eelbo (2003)*. The user can work autonomously with the information at hand. After finishing work, the work items are re-synchronized with the central information portal. Synchronization of personal and central information portals is done transparently to the user. Different conflict resolution strategies can be chosen, e.g. automatic conflict resolution and interactive conflict resolution.

As an example, a surveyor can browse a task list on his personal information portal. Tasks assigned to him are transparently synchronized to his personal information portal. Work-related artifacts like survey items, survey descriptions, ship information, etc. are made available as well. Completed surveys are published to the corporate information portal. The conflict resolution strategy chosen for this scenario determines that a member of the society head-office accepts or rejects proposed changes submitted by the surveyors, e.g. ship information updates.



Fig.5: Personal portals providing local work contexts synchronized with the central portal

3. Mobile Services in the Maritime Industry

In maritime industry IT support of stationary workplaces is prevalent. Stationary workplaces are office desktop computers, but also movable devices, e.g. laptops. Portable or mobile devices, previously only used as organizers and phones, start to become full-fledged work devices. With the enhancement of device capabilities of smart phones and PDAs that now comprise of multimedia capturing and processing means, these devices are ready to be introduced to maritime industry's processes.

We see mobile services in maritime industry especially suitable in on-the-spot documentation processes and providing information services, *Aziz (2003)*. The requirements for mobile services in maritime industry differ from those of mobile services developed for private use (e.g. finding points of interest; navigation systems in automotive industry): Surveyors performing a survey on a ship are not permanently connected; the wireless networks are not always available, e.g. in a ship's engine room. Therefore, surveyors must be provided with an offline work environment through their mobile device. Mobile services are also suitable for ship operators who work in their offices but are very mobile, e.g. attend ship survey, visit shipyards, etc. They require up-to-date information wherever they are even without permanent Internet connection.

In this research project, two prototypical mobile services have been developed: A mobile survey service supporting surveyors during inspection and a mobile office finder service for society customers and staff have been developed. The mobile services connect to the service portal using WebServices, *Ge (2003)*.

3.1 Mobile Survey Support

The mobile survey support facilitates the existing ship survey process. The service relieves surveyors from visiting one location several times by grouping survey items by position on ship (items on deck, items in the engine room, etc.). The survey process is furthermore improved by making use of mobile

device multimedia capabilities: Surveyors can additionally document survey items with images and audio comments.

A typical survey scenario is given below, Fig.6. At the port office, a surveyor loads the survey item list from his personal information portal onto his mobile device. He disconnects from the portal and starts performing the survey onboard a ship. During this survey, he checks individual survey items and records their state. He may add a short notice, enter a measurement or date. Additionally, he may document the survey item state with a picture or record an audio comment for later use when writing the survey report. He may recall information collected during the previous survey. This allows him to compare e.g. the current state of a bulkhead with its state at a previous survey to track corrosion development. In case a surveyor cannot complete a survey, the mobile service allows him to delegate certain parts of the survey among the local community of surveyors. At the end of a survey, the results are published to the personal information portal and can be edited. The resulting report includes state information, notes and pictures from the survey. The survey information together with the report is published to the classification society's central portal (see section 2.3, synchronization of portals, for more detail).

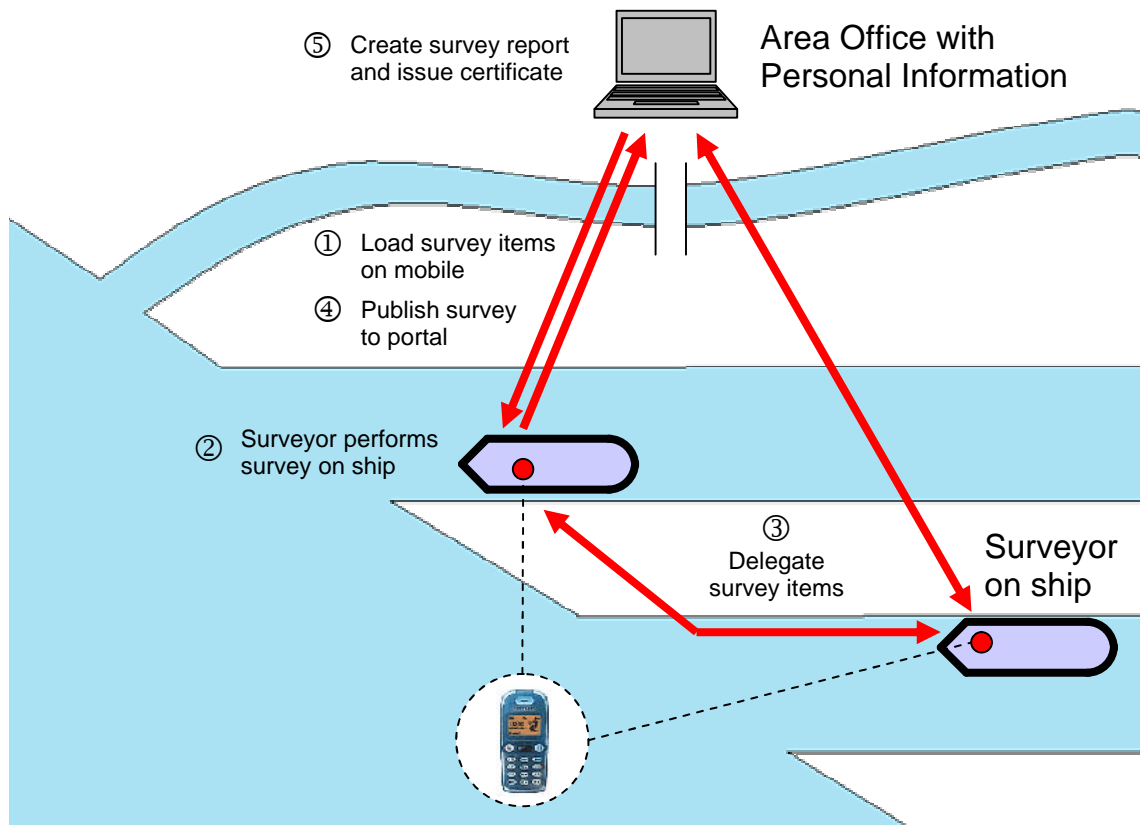


Fig.6: Mobility Support for Ship Survey

The mobile survey support has been realized as a prototypical mobile service on a Nokia Series 60 cell phone. Despite of the technical limitations of current devices concerning multimedia capturing and Java platform support, the service can improve the ship survey process.

3.2 Mobile Office Finder

The mobile office finder application allows customers and society staff to locate society offices in a region or country and to contact office personnel by call or by sending SMS messages. For this, contact information already available on the Internet and as a printed handbook have been made available as a mobile service.

The basic contact information provision service has been extended with a location-based service, which assists the user in finding the nearest office based on his current location, Fig.7: As a first step, the user's current position is determined. The user can enter the name of the city in which he is currently residing. Other options include the localization via postal code or using GPS positioning.

An essential factor is a supportive user interface. A user might not know the correct spelling of city names (e.g., Marseille, Edinburgh). Phonetic matching of user input with the city names is therefore useful. If a name cannot be matched exactly, the user is presented a list of phonetically near city names, together with additional information. One can think of postal code or geographical location within the country. The user can choose from the list or refine his city name input. Once a city is selected, the service can easily determine the nearest port office and provide the user with contact information. He can contact the port office by phoning directly or by sending messages.

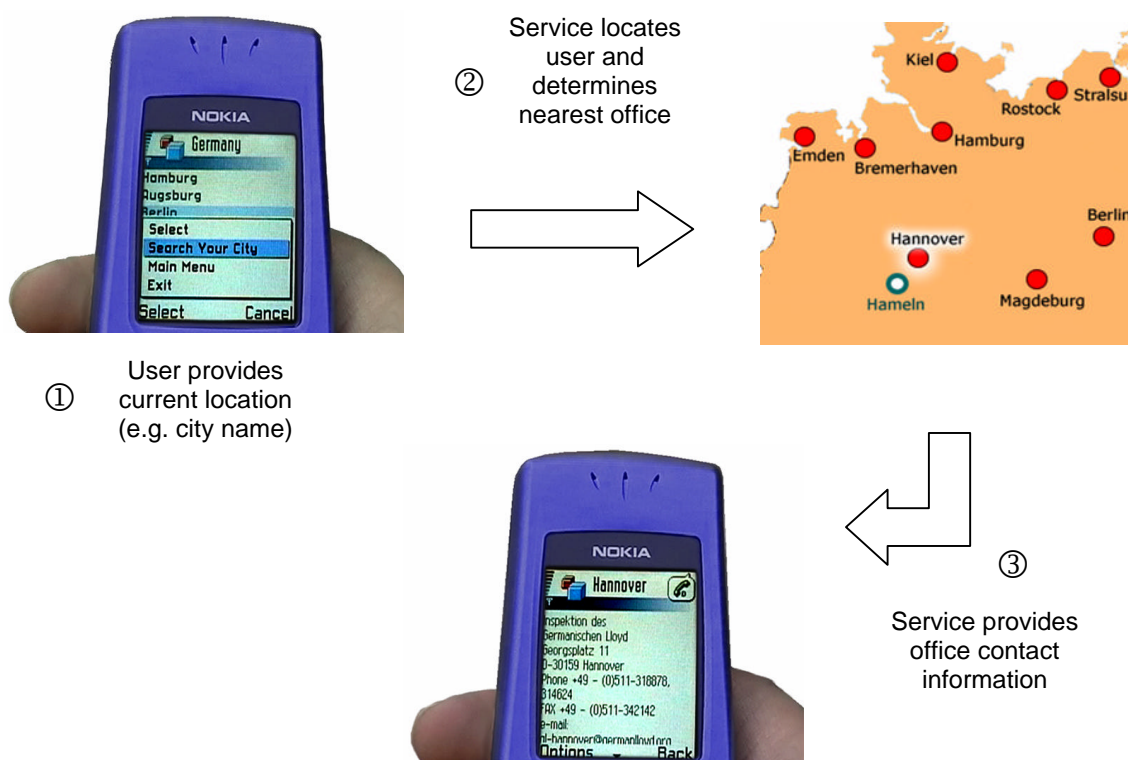


Fig.7: Mobile Office Finder Service determining the nearest office and displaying contact information

The two mobile services were realized on *Java 2 Micro Edition* runtime platform, focusing on connected mobile devices, i.e. cell phones and personal digital assistants (PDAs). The advantage of using Java as the runtime platform - instead of *Pocket PC* or the *Symbian* operating system - is that services can be deployed to any mobile device complying with the mobile Java runtime platform. Deployment of services is very simple for cell phones, where the *Java Over-The-Air (OTA)* deployment mechanism can be used, Yuan (2003). Deployment of services on non-connected PDAs requires the PDA owner to perform the service installation.

Practical experience showed limitations of the chosen technical platform: The functionality offered through the Java standardized interfaces and via optional Java packages is currently only a subset of device functionality available via native programming interfaces. Still, the advantage of having a common service platform for mobile devices like smart phones and PDAs weighs more than the observed limitations.

4. Future Work

The mobile services presented can be extended in future work. The mobile survey support service (see section 3.1) can be extended to optimize the planning and coordination of ship surveys. This includes synchronization of a survey conducted by various surveyors. Currently, only preliminary distribution of surveys to multiple surveyors is supported. The survey items are assigned and loaded onto the mobile device of each surveyor depending on the ship area he is in charge. Later on, this process can be further optimized by not only loading survey items relevant to surveyors' position but also optimizing the process path taken by various surveyors present on the ship to save effort and time. This service can be particularly helpful due to different design of individual container ships.

The abovementioned service may pose two major difficulties. One is the download and storing of data onto the mobile device for working in the ship parts where there is no connection available, either to the personal portal or to the Internet, and the second important question is regarding the processing of large amount of ship design data needed to compute the optimal path for any particular survey. The first problem requires research in the area of structuring multimedia data on the mobile device for efficient memory use as well synchronization with the personal portal. The second problem has more to do with the limited processing power available for the mobile devices. The research area can be divided into two main parts:

- To probe the possibility and suitability of a distributed work environment for mobile devices, where parts of the complex processing can be performed on more powerful devices and then results conveyed to the mobile device which does local processing with the help of additional information.
- Effort to determine the relative difference between the capabilities of mobile devices and their relatively less mobile counter parts.

The Mobile Office Finder service (see section 3.2) currently allows finding the nearest office with respect to geographical distance. While this is suitable in Europe where multimodal transportation systems (rail, road, flight, ship transport) have a dense covering, it might be more appropriate for other regions to enhance the service to locate the nearest port office with respect to available routes and transportation means. Furthermore, route planning services for multimodal transport could be integrated to complete the Mobile Office Finder service.

In the long run, we want to realize personal information portals on mobile devices that enable users to handle relevant services and information from their personal information manager. Mobile device capabilities are expected to increase and to allow such *mobile personal information portals*.

5. Summary

In this paper we report on the results of a joint national research project undertaken by the Software Systems Institute research institute of TUHH and Germanischer Lloyd.

We report on an online service portal for a classification society, its partners and associated operators, i.e. ship operators and surveyors. To provide an encompassing solution our project focuses on the following contributions:

- Ontological structures for information and knowledge to help users retrieve ship survey information, to find related information and contact persons;
- central access points to an example suite of services tailored to users needs for a range of time-critical tasks from ship construction to ship operation; and
- a generic software platform for business processes involving a classification society and its business partners in different modalities.

The experience gained from our combined approach to information and knowledge management, service integration and process support via online service portals has laid grounds for building generic

mobile and location-based services. Ship owners, classification society staff and surveyors can use the proposed service infrastructure in a wide variety of working contexts including mobile and stationary work environments, online and offline working modes and working platforms ranging from desktop computers to laptops, PDAs and smart phones. Prototypes of mobile services have been developed demonstrating the feasibility of the chosen approach.

Acknowledgement

The work presented in this paper is supported by the German Federal Ministry of Education and Research (bmb+f) under grants EUT 871a (WIPS IT 1.1) and 03SX166 (WIPS IT 1.2).

References

AZIZ, M. (2003): *A model for mobility-aware information system services*, Master Thesis, TU Hamburg-Harburg

CARZANIGA, A.; FUGGETTA, A.; HALL, R.S.; HEIMBIGNER, D.; HOEK, A. van der ; WOLF, A.L. (1998), *A characterization framework for software deployment technologies*, Technical Report CU-CS-857-98, Dept. of Computer Science, Univ. of Colorado

EELBO, J. (2003), *Synchronisation von Besichtigungsinformationen (Synchronization of survey information)*, Studienarbeit, Technische Universität Hamburg-Harburg.

EURIFT (2003), *European Research Center for Intermodal Freight Transport*, Research Project Website, <http://www.eurift.net/>

infoAsset AG (2001), *The infoAsset Broker*, Technical White Paper, www.infoasset.de

GE, J. (2003), *A webservices-based interface for an information portal platform*, Diplomarbeit, TU Hamburg-Harburg.

GL (2004), *Rules for Classification and Construction*, I – Ship Technology, 1 – Seagoing Ships, 2 – Machinery Installations, Germanischer Lloyd, Hamburg, <http://www.gl-group.com>

HUPE, P. (2000), *Eine daten- und prozessorientierte Architektur zur Integration kooperierender Informationssysteme (A data- and process-oriented architecture for cooperative information systems integration)*, Diplomarbeit, Univ. Hamburg.

Jagoe, A. (2003), *Mobile Location Services*, Prentice Hall.

LEHEL, V. (2002), *Synchronisation von Informationsobjekten zwischen Portalen (Inter-portal synchronisation of information objects)*, Diplomarbeit, TU Hamburg-Harburg.

WEGNER, H. (2002), *Analyse und objektorientierter Entwurf eines integrierten Portalsystems für das Wissensmanagement (Analysis and object-oriented design of an integrated portal system for knowledge management)*, Dissertation, TU Hamburg-Harburg.

HUPE, P.; SCHMIDT, J.W. (2004), *WIPS IT 1.1 - Internet Services for Ship Technology*, Research Project Report, TU Hamburg-Harburg.

LANGBECKER, U. (2004), *WIPS IT 1.2 - Internet Services for Ship Technology*, Research Project Report, Germanischer Lloyd.

YUAN, M. (2003), *Enterprise J2ME - Developing Mobile Java Applications*, Prentice Hall

FEM Supported Alignment of Power Train Components

Jan Henrik Weyhardt, Berend Bohlmann, Flensburger Schiffbaugesellschaft GmbH &Co. KG,
Flensburg/Germany, Weyhardt@FSG-ship.de , Bohlmann@FSG-ship.de

Abstract

The paper presents lessons learnt from FEM computations on the interaction of the aft body and the gear boxes of a RoPax ferry. The calculations were conducted due to insufficient tooth grip pattern which were observed at trial trip conditions. Other examples on the deflection behavior of modern RoRo vessels, powered conventionally by 4-stroke plants on gear boxes as well as gearless 2-stroke plants address some aspects on assembly modes and operational conditions.

1. Introduction

Ship yards, engine builders and gear box makers put great efforts to increase the efficiency of their products. The steel weight of modern ship hulls is significantly reduced compared to former generations. Hull form and compartmentation are much more efficient and this process is still ongoing, thus allowing the yards to improve their product performance in terms of payload capacity, speed performance, fuel consumption etc. Analogously, this applies for the suppliers of propulsion trains: Engines, gearboxes, clutches and bearings generate and carry higher power densities than ever.

1.1. General problems

Unfortunately, the progress in overall power/mass efficiency has in certain cases resulted in insufficient power/stiffness relations of components which in turn affects the whole system. Beside the conventional assessment of the single components, the interaction of the hull and the power train components is to be evaluated. However, local weaknesses are not generally of disturbing influence, but can be advantageous for a working alignment. Another point is that mechanical engineering components are quite sensitive against misalignment and because of an insufficient knowledge about interactions to hull close tolerances are predicted to avoid failures.

1.2. Definition of sufficient alignment

“The alignment of a propulsion train is carried out while a berth and/or building state (irrelevant for service) in a way, that no combination of loading, ballasted and swell conditions, propulsion parameters, local temperatures etc. (relevant for service) leads to irregular states for bearings and (crank) shaft.”

This alignment definition is “permanent under construction” and has to be fitted to each special application by weighting the single items. It does not demand an optimal alignment for the design load case because the changes need not to be distributed symmetrically around it. The definition can include the regarding of states corresponding to the likelihood of appearance (service strength). It is intended to allow additional bendings to enlarge alignment tolerances. Another point of “sufficient” is the realization of calculation and modelling within a tenable time!

1.3. Goals

It has to be find a far-reaching knowledge of the interactions between shipbuilding and mechanical engineering components to enlarge alignment tolerances and reliability. This knowledge can also be used to create revised or even new building methods e.g. alignment on building place before launching or before locating major block weights and/or pre-outfitting of blocks with propulsion train components in preassembly stages.

2. Correction of a RoPax ferry's gear box alignment

The example is suitable for the interactions between shipbuilding and any mechanical engineering components, like gear boxes, main and auxiliary engines, PTOs etc. It deals with a twin-screw RoPax ferry, powered by 2x2 4-stroke engines with a total power of about 22000kW, Fig.1.

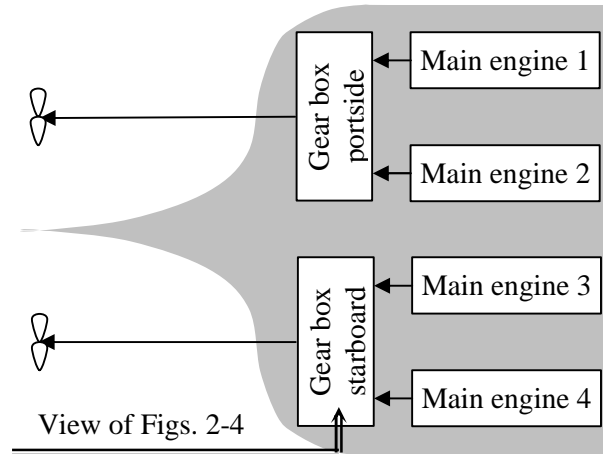


Fig.1: Plant arrangement in aft ship

2.1. Problem

The alignment of the gear boxes had to be corrected due to insufficient tooth grip pattern on all four engagements of driving pinions and following gears, which were observed at trial trip conditions, Fig.2. It has to be taken into account that after some hours of service conditions a tooth grip pattern is in state of development. Nevertheless, at the first sight the foundation seemed to be too weak. This point of view was emphasised by the fact that this vessel was weight critical while construction and as a result there were some rumours that the foundations of the gear boxes appeared too weak. However, a point of weakening gear foundations is the fact, that the following wheel's big diameter demands a cut-out in the upper deck of double bottom, directly on the fore side of the thrust bearing. FSG was asked to evaluate the influence of foundation to the tooth grip patterns and for suggestions to improve them to everyone's satisfaction.



Fig.2: Insufficient tooth grip pattern on starboard gear box, view as shown in Fig.1. In the foreground the starboard driving pinion, in the background the following gear

2.2. Causes

First researches resulted in a more complex problem: In the early design stage the gear supplier thought FEM modelling and calculations to the gear itself are unusual. On the other hand he demanded from the yard to guarantee a maximum gear shafts crossing of 0.1mm/m without recognizing the gear house stiffness. So the yard took a “dummy gear” FEM calculation to design a light weight foundation with sufficient stiffness. Obviously there was no more relevant communication. Each partner designed his own part as shown in Fig.3 on the final FEM model, realized after trial trip.

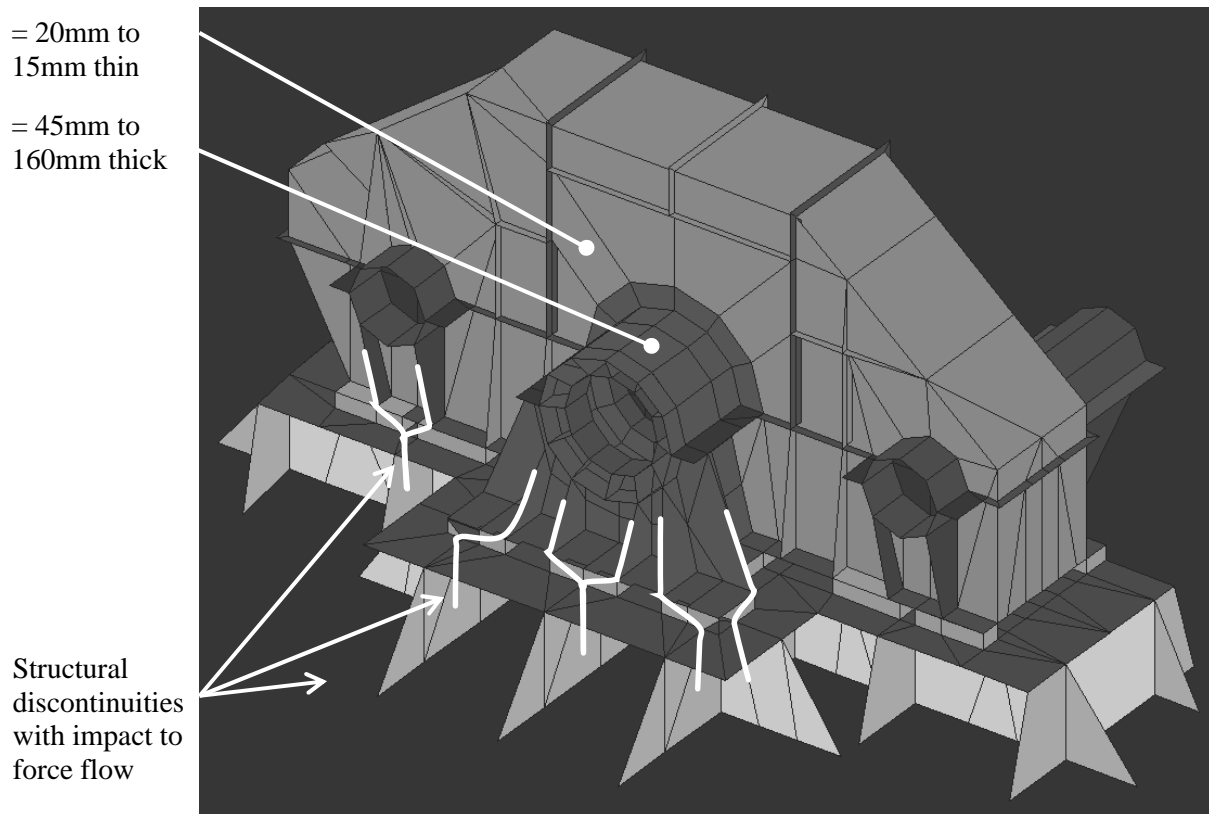


Fig.3: Disregarded basic design rules at the cut of gear box and foundation:

The gear box is coupled to two engines on the back side of view and the shaft line on the fore side. There is also placed the dark grey coloured thrust bearing. The following basic design rules were disregarded:

- No needless jump in material thickness (the thinner, the lighter drawn). No material of 20mm to 45mm is used. The thrust bearing is stiff against environment.
- No needless structural discontinuities with impact to force flow at the transition of gear and foundation.

Conclusion → more stiffness was possible with less material.

2.3. FEM-calculations

The FEM model shown above is necessary to calculate the actual deformations in the system of interacting gear box and foundation. Bearing stiffnesses, shafts and engaged gear wheels are not necessary to evaluate the quality of foundation. According to the service condition “both engines full power” the model was loaded with the maximum thrust and bearing loads resulting from maximum torque. The deformation behaviour can be explained by the vertical displacements shown in Fig.4.

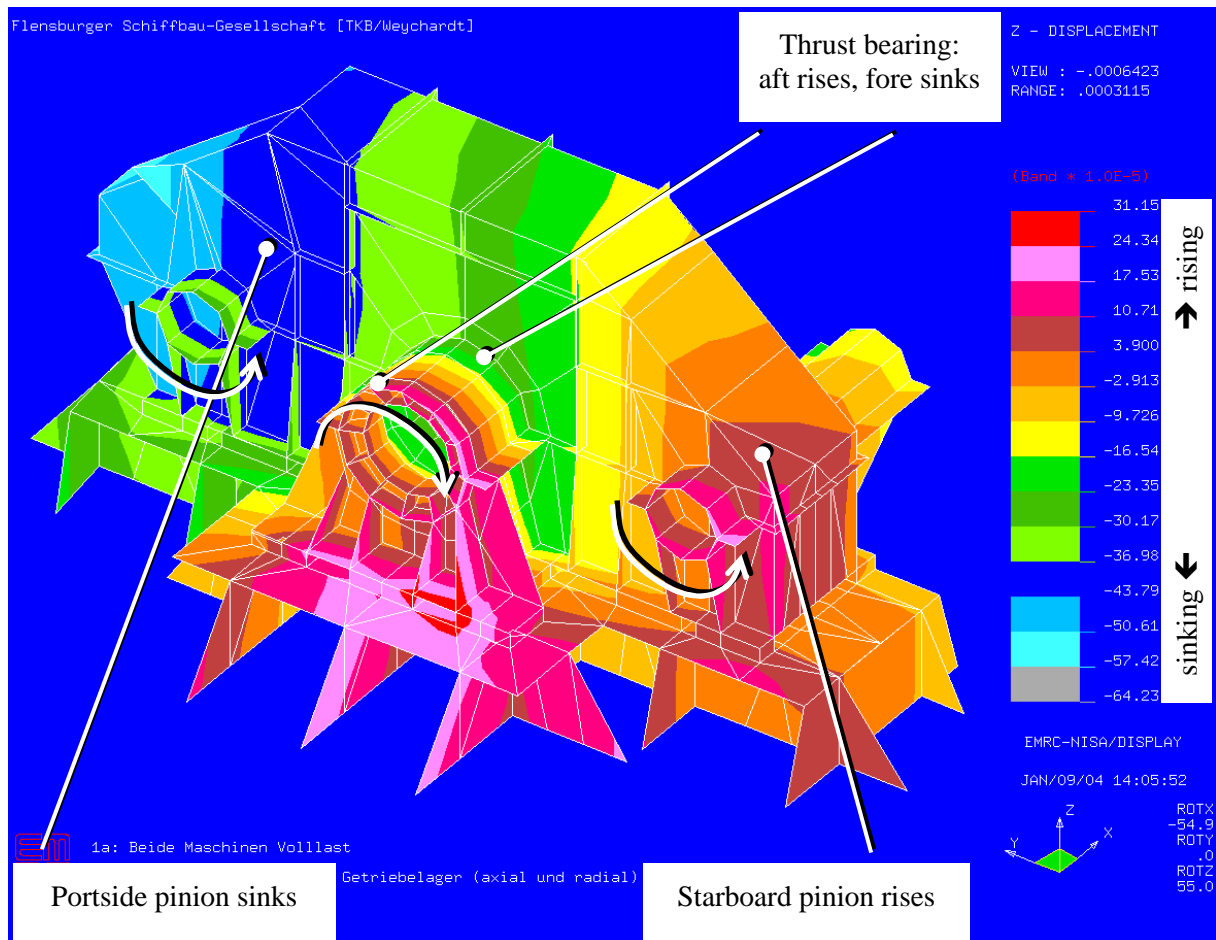


Fig.4: Vertical displacements at portside gear, propeller rotates clockwise, engines counter-clockwise, sinking and rising of pinions due to tooth force directions

The sinking of the portside pinion can easily be detected by the darkest grey colour in grey scale. It sinks because the downwards directed reaction force on the following gear due the counterclockwise rotating engines. Persecuting the grey scale on the housing to the rising starboard pinion indicates a smooth tilting of the whole gear box. A much stronger tilting can be detected at the thrust bearing with more slender grey colour stripes.

The relevant results were:

- Crossings of the shaft's projection lines on the top plate of foundation between shaft's bearings: $0.04\text{mm/m} < 0.1\text{mm/m}$. Result: The structural stiffness of the vessel is sufficient.
- Crossings of the shaft's lines between shaft's bearings in gear house: $0.14\text{mm/m} > 0.1\text{mm/m}$, the gear house's part is 0.1mm/m and does not fulfil the demand of its own manufacturer. The calculated crossing will even increase by taking into account bearing play and weakness of shafts, gear wheels and tooth engagement.
- Tilting of thrust bearing to cross axis (not demanded): 0.4mm/m , as consequence the shaft tilts itself and moves up in the thrust bearing.

Other asymmetrical load cases caused smaller crossings.

2.4. Problem solution

A new shaft alignment towards the gear box was calculated. Some journal bearings had to be raised. After the second trial trip the tooth grip patterns were almost satisfying, Fig.5. Another optimisation was carried out by the gear manufacturer by smooth tensings of the housing.

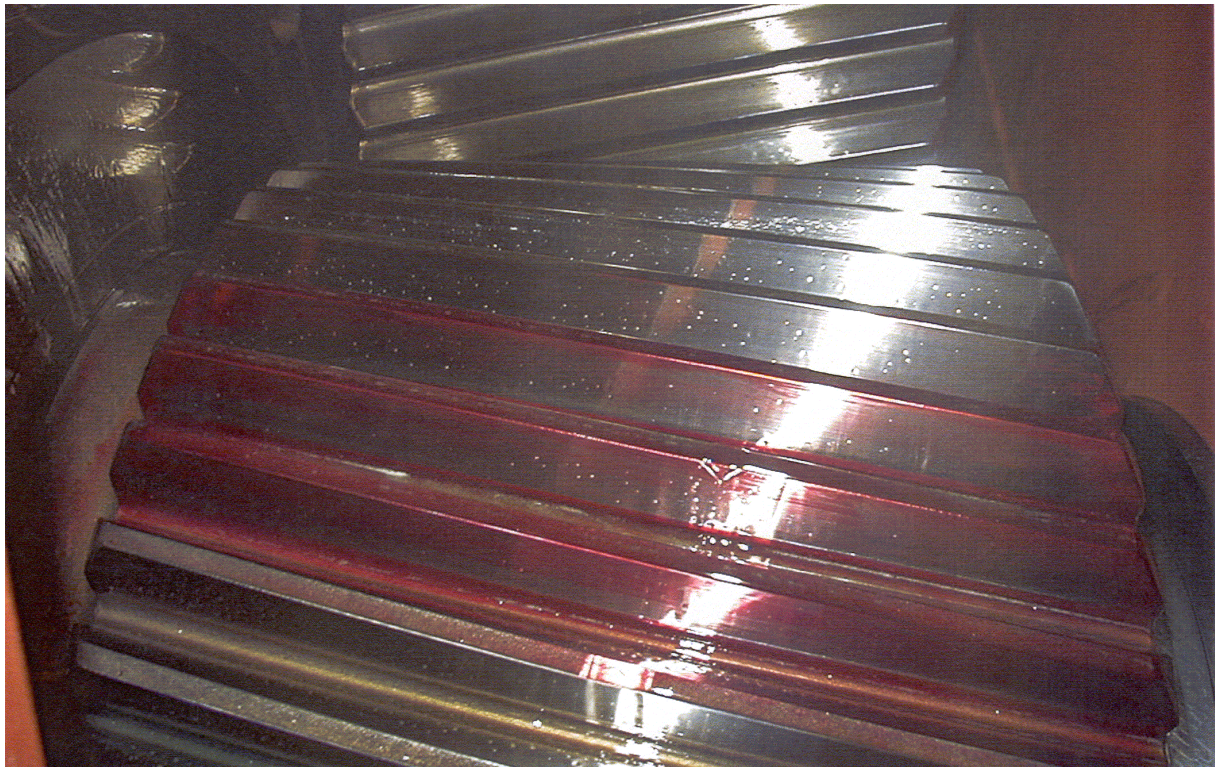


Fig.5: Sufficient tooth grip pattern, same pinion/wheel engagement as in Fig.2

2.5. Lessons to be learnt

Costly amendments can be avoided by using available technologies in early design stage. In this case the FEM-modelling of the gear box, its foundation and the close environment would have been sufficient. To recognize basic design rules an approach of mechanical and ship building engineers is necessary regarding the fact that supply components only can be reasonable in series production. That is only possible with a frictionless communication.

FEM calculations in early design stage can be a stable base for supplier guarantees and for the yard a help to find the best fitting component. A close attendance by classification society is desirable. Such calculations also can avoid illogical demands like a maximum shaft crossing without modelling the shafts and their environment. In this case the quality of tooth grip should have been demanded because it was the reason of query.

3. Crank web clearance calculations by FEM

FSG carried out some studies referring the crank web clearance of 4-stroke 9-cyl. engines, because it is one of the criteria for a sufficient alignment of main engine. The crank web clearance is the distance change of two shaft segments neighbouring to the same crank at one revolution. The amplitude can change at different service conditions, e.g. caused by thermic deformations. So the interaction of crank shaft, engine and double bottom is interesting to be evaluated.

A sufficient modelling of double bottom's and engine's steel structures and lubricating films is state of the art, see Ch.4.1. However, it is not a standard computation at yards, classification societies and engine makers and even in special cases rarely used until today. The emphasis was laid to the crank shaft with its unsteady geometry. Two starting points were discussed: The modelling with beam elements or brick elements.

Beam elements are easy to define and quick to calculate. But a model reflecting all the mechanical behaviour at tensile, bend and torque loads was not found with a tenable expenditure. Further computation rates increased so that big models can be calculated in a sufficient time. So a model of one crank segment was made out of brick elements, Fig.6. A crankshaft can be modelled by connecting any number of these segments at any angles by rigid links at the marked centre point. So it is not necessary to adapt the meshing on the end wall to different angles resulting from the number of cylinders and/or the ignition sequence. Most certainly other inaccuracies in modelling cause bigger deviations. However, a linear deflection behaviour is assumed for the transition which might give reasons for further discussions.

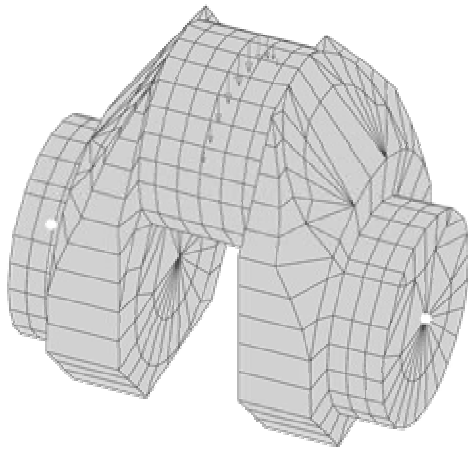


Fig.6: Crank segment for one cylinder made of brick elements

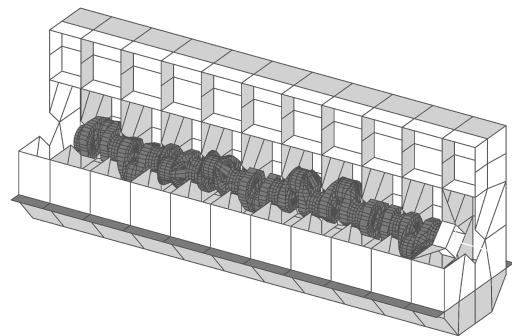


Fig.7: Crank shaft made of 9 segments in engine housing

A 9-cyl. crank shaft embedded in the engine housing is shown in Fig.7. This ensemble was connected to a double bottom structure and bended by a reference moment. The FEM-model of crankshaft was revolved in 40° steps for each of 9 calculations and the crank web clearance was determined for each cylinder. The results are sufficient, Fig.8. All curves are nearly sinusoidal as expected. The amplitudes depend on the relative angle between two cranks. Small amplitudes are calculated for the cylinders no 3, 4, 6 and 7. The reason is that the relative angles from no 3 to 4 and 6 to 7 is 80° instead of 40° for all other neighbouring cylinders.

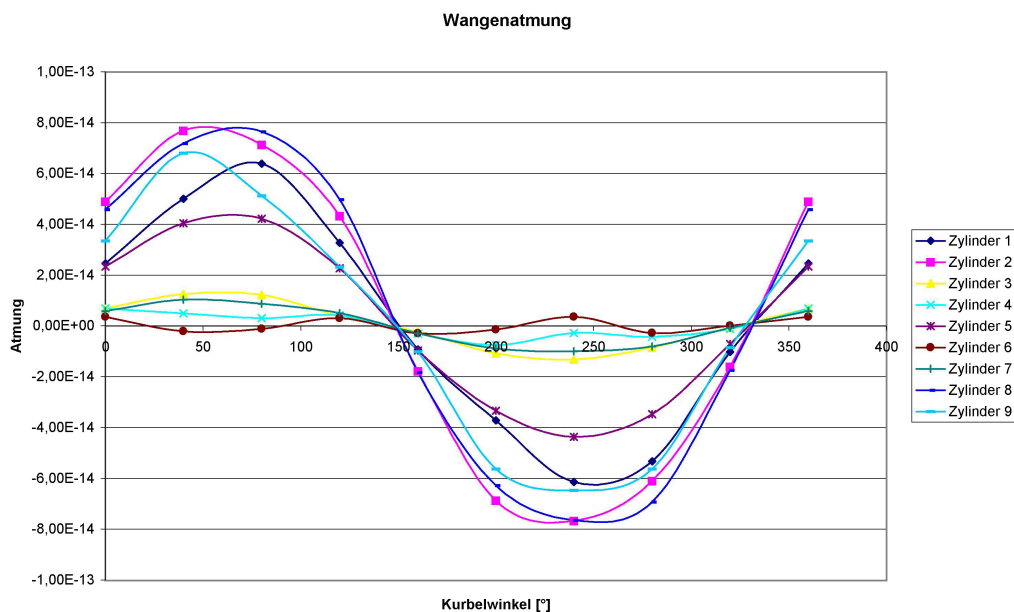


Fig.8: FEM calculated crank web clearances at 9 cylinders

4. Alignment of a RoRo vessel's propulsion train

At present FSG builds 5 RoRo vessels with a FEM supported alignment. Bearing displacements resulting from hull deformations of all relevant building, outfitting and service cases were regarded in bending line calculations.

4.1. FEM models

All calculations for different cases were carried out with variants of a global FEM model taken from the vibration analysis. The meshing in the lower aft ship area had to be refined, Fig.9.

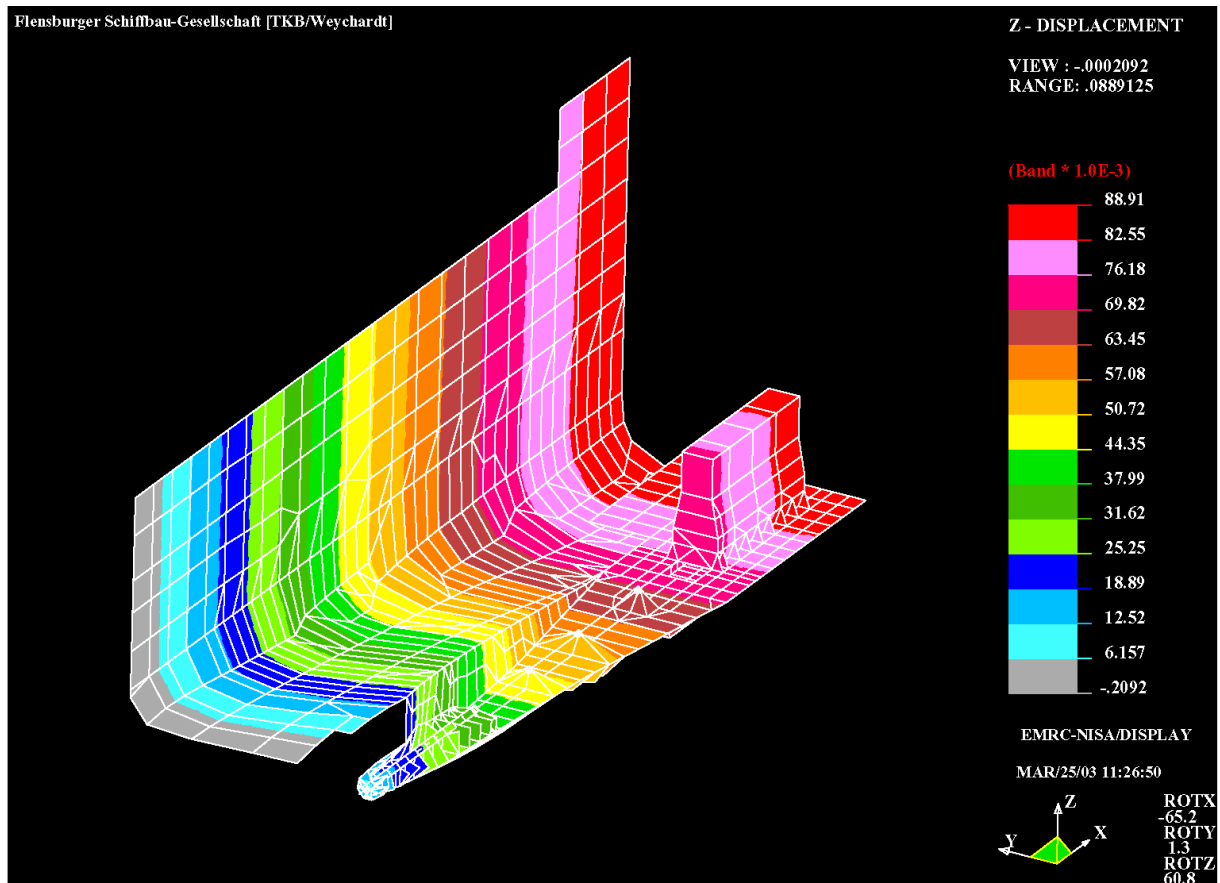


Fig.9: Cut out of global FEM model showing aft ship (shell) with refined meshes, main engine and shaft line

The steel structure was modelled by shells and beams with linear deflection approaches and varies with the building state. Individual mass distributions are realized by single masses on each node. There are also inducted case specific service forces. The water pressure distribution regarding draught, trim and heel is inducted normal to dipped shell elements. Shaft and hull are connected by springs representing the stiffness of lubricating films in each bearing at specific service conditions.

4.2. FEM calculations

As all results, the calculated displacements in Fig.9 refer to an imaginary state without any deformations. So they are not suited for conclusions and have to be referenced to other deformation cases. In this application a reference deformation case “according to specification” (typical loading conditions simulating cargo load, bunker etc. according to owners requirements) for best alignment could be defined, because the main changes resulting from maximum wave sagging and hogging are nearly symmetrical and the different load cases cause only negligible changes. Some building and

outfitting states had to be regarded to calculate the changes from alignment states to service state. About 20 deformation cases were defined, e.g. to evaluate changing conditions while building or outfitting. The most important were:

1. According to specification (still-water service conditions)
2. Maximum hogging
3. Maximum sagging
4. Outfitting alongside quay (alignment finish)
5. Assembling aft ship block (first alignment under slipway conditions, w/o main engine and deck house)

Deformation changes inside propulsion train can be evaluated, if all calculated displacements are referenced to any two points, e.g. aft stern tube bearing and fore end of engine, Fig.10. The first three graphs in agenda show the vessel's deflections caused by the maximum vertical design wave bending moment according to class, e.g. the maximum change between maximum hogging and sagging ("3-1") is about 5mm. The other three give information, which displacements have to be provided while alignment to reach the best result in service conditions, e.g. the influence of launching is about 3mm ("5-4").

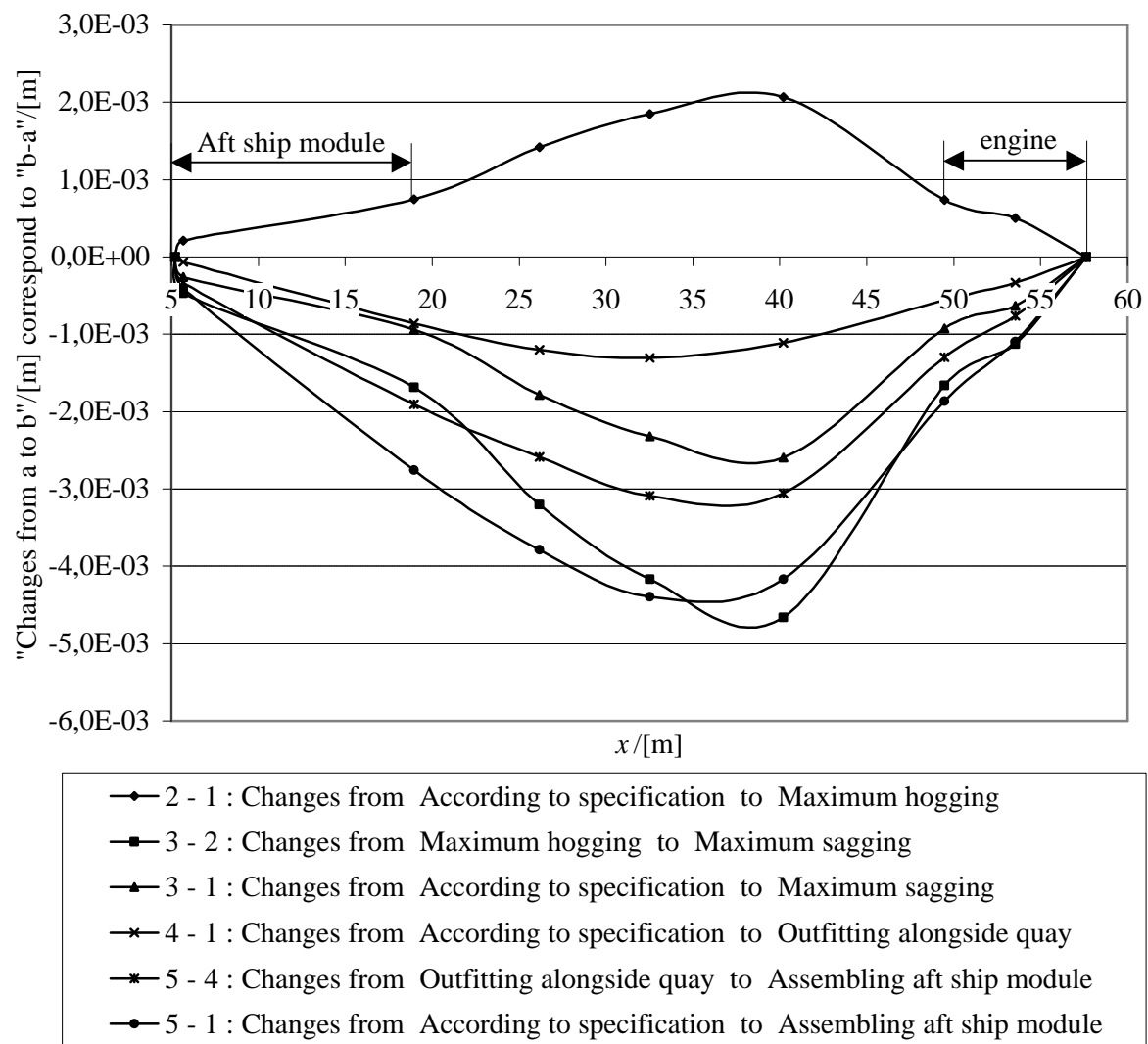


Fig.10: Deformation behavior (vertical) of a RoRo vessel's aft ship structure changing between different deformation cases

4.3. Calibration/Validation of the FEM model

The most sensitive location against misalignment in the propulsion train is the shaft bossing on the aft ships block, Fig.11. It is designed for optimal propeller inflow, so the lever arm from aft stern to the 35t propeller is approx. 6.5m. The decision to calibrate the FEM model here was founded in three reasons:

- All FEM calculations of vessel show the most significant bending in this region of lowest stiffness, Fig.10,
- the calculated alignment value for the aft stern tube bearing of about 15mm above center line is unusual high and
- a damage of this bearing is most costly because of docking.

The bending behavior was tested with a beam, welded in two points of 8m distance under the block and a weight force on the aft end, Fig.11. The weight force bends only the shaft bossing but not the quasi elastic linked test beam.

So the relative deformations were measured without boundary effects and could be compared by a FEM calculation. The requirements to the weight were:

- At least 20t to cause measurable deformations (pre-calculated),
- easy to transport and
- height incl. lifting device less then 3.5m to fit under test beam.

The choice was the elevating platform truck that normally carries ship blocks. To be carried by a block as shown in Fig.12 is a special load case that had to be approved by the manufacturer.

Fig.13 shows the good agreement between calculation and measurement. A last validation of the unusual alignment values are successful trail trips and service performance.



Fig.11: Aft ships module with shaft bossing, test beam and weight force F

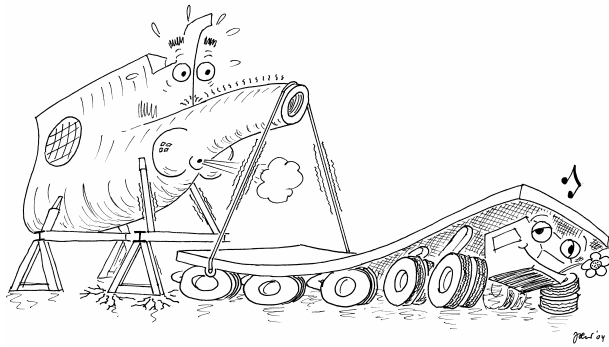


Fig.12: Topsy-turvy world:
Module carries elevating platform truck

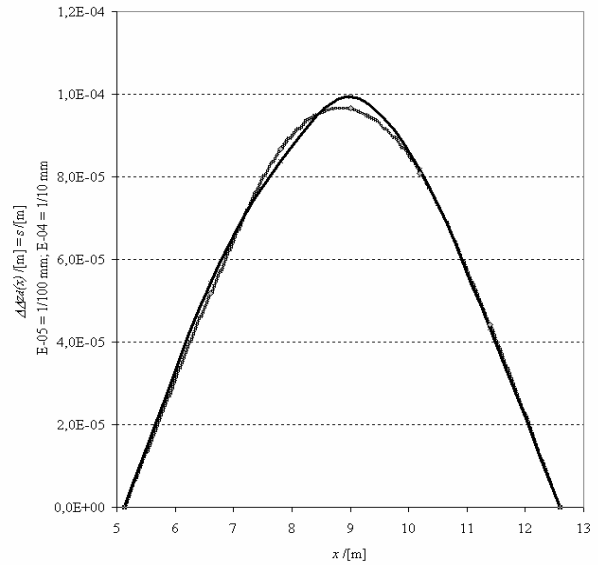


Fig.13: Comparison calculation-measurement
converted to a weight of 10t,
FEM (grey), measurement (black)

References

BOURCEAU, C.; WOJCIK, Z.C. (1966), *Die Beanspruchung von Kurbelwellen in Dieselmotoren*, Jahrbuch der Schiffbautechnischen Gesellschaft

CASTLE, D., (1969-1979), *Alignment Investigation following a Medium-Speed Marine Engine Crank Shaft Failure*, Proceedings of The Institution of Mechanical Engineers

DeGEORGE, V.A. (1982), *Combined Effects of vertical and horizontal Shaft Alignment on Main Reduction Gear*, Marine Technology, pp.178-184

ILLIES, K. (1969), *Wechselwirkungen zwischen Maschine und Schiff*, Jahrbuch der Schiffbautechnischen Gesellschaft

N.N. (2000), *Rigid shafting and flexible hulls*, The Motor Ship, pp.87-92

Optimisation of the Survivability of Naval Ships by Genetic Algorithms

Evangelos K. Boulougouris, NTUA-SDL, Athens/Greece, vboulg@deslab.ntua.gr

Apostolos D. Papanikolaou, NTUA-SDL, Athens/Greece, papa@deslab.ntua.gr

Abstract

Surface warships are a special category of ships given the fact that compared to other ship types they are designed to operate in a lethal environment. Therefore their survivability is a vital feature of their design. The damage stability characteristics form a major design parameter of the vulnerability and therefore of the survivability of a naval combatant. The naval architect is asked to minimize the vulnerability of the surface warship through optimal watertight compartmentation considering a variety of damage scenarios and operational/environmental conditions. On the other hand there are several restrictions imposed by the space requirements for the accommodation of the various weapon and other ship's vital systems (e.g. naval guns, vertical missile launchers, power, command and navigational units etc) but also by the basic requirement for the minimization of the weight of the steel structure of the ship. This forms eventually a very difficult design problem where multiple objectives are set and multiple constraints have to be met. Using the Multi-Objective Genetic Algorithms (MOGA) this optimisation problem can be solved very efficiently. The herein implementation of the method integrates the well-known ship design software package NAPA with the general optimisation software package modeFRONTIER. The paper analyses the results of a case study optimisation on the basis of the hull form of a modern naval combatant.

1. Introduction

Compared to other ship types a unique requirement for naval ships is that they must operate adequately in a lethal environment. Therefore their survivability is a vital design objective. The damage stability and floatation characteristics play a determinant role for the vulnerability of naval vessels and have a major impact on their survivability. The naval architect is asked to minimize the vulnerability of a surface naval ship through optimal watertight compartmentation considering a variety of damage scenarios and operational/environmental conditions. On the other hand there are several constraints imposed by the space requirements for the accommodation of the various weapons and other ship's vital systems (e.g. powering, command and navigational units, various weapons, vertical missile launchers, etc) but also the basic requirement for the minimization of the ship's structural weight. Moreover, modern naval warfare is characterized by highly sophisticated weapon systems. Surface combatants are threatened by air, surface and underwater weapons guided with various sensors: radar, infrared, electroptic, laser or acoustic. In order to accomplish their mission they have to carry a large arsenal and a complicated suite of advanced (but nevertheless sensitive) electronics. These have increased their acquisition and operational cost and reduced the size of the fleets operated by various Navies. Then again the need for high payload to displacement ratio has driven the designers to a reduction of the shell plate thickness for keeping the structural weight as low as possible. This resulted to a shift *from armour to sensor capability*, making naval ship designs more vulnerable. It becomes obvious that an effective solution to this problem is the adoption of a rather new design philosophy, namely *Design for Enhanced Survivability*.

The above led to the introduction of *survivability* as one of the basic requirements for the design of modern naval ships. For instance, the U.S. Navy has published a number of documents explaining its strategy and survivability policy (Said, 1995):

- OPNAV Instruction 9070.1, establishing "survivability [as] a fundamental design requirement of equal significance to other inherent ship characteristics. It shall be accorded the same consideration in design as other basic elements."
- DOD Instruction 5000.2, designating survivability considerations that must be addressed at each milestone decision point throughout the acquisition process required for the Defence Acquisition Board (DAB).

Moreover, several authors have proposed a direct link between the operational requirements and the vulnerability of naval ships in order to rationalize their design process, *Reese et al. (1998)* and *Alman et al. (1999)*. The problem for the naval architect is, as could be expected, that he has to consider already in the early design process a series of conflicting requirements. This results to a design solution space that is very large, non-linear, discontinuous and bounded by a variety of constraints and thresholds, *Brown and Thomas (1998)*. Therefore there is a need for a methodology that quantifies the probability of survival in a systematic way, giving the designer a metric to investigate the impact of his decisions on the survival capability. This would also permit the introduction of survivability as an objective into the multi-objective design optimisation problem and not as a constraint, *Vassalos (1999)*.

2. Naval ship survivability

Survivability may be defined as “the capability of a ship and its shipboard systems to avoid and withstand a weapons effects environment without sustaining impairment of their ability to accomplish designated missions” (*Said, 1995*). It contains two aspects:

- The *susceptibility* defined as the inability to avoid being damaged in the pursuit of its mission and to its probability of being hit (P_H).
- The *vulnerability* defined as the inability to withstand damage mechanisms from one or more hits and the probability of serious damage or loss when hit by threat weapons (P_{KH}).

Thus mathematically *survivability* (P_S) is related to the other two quantities by the following formula, *Ball and Calvano (1994)*:

$$P_S = 1 - (P_H \times P_{KH}) \quad (1)$$

It should be considered that a naval ship has the capability to restore at least a part of the damage incurred through her damage control procedures. This has introduced the concept of the *recoverability*, which involves the actions taken by the damage control parties to restore damaged systems and enable ship's operation at a level higher than that immediately after the hit. By definition recoverability is mainly an operational aspect relying on the sufficient training of the crew although it may still pose several requirements to the designer. This aspect will not be addressed within the survivability model presented herein.

Operational aspects affect the susceptibility of a naval ship, but the major influence factor is due to the intrinsic characteristics of the vessel, namely its signatures. Electronic emissions such as radar scans or external communication attempts could reveal the position of the stealthiest vessel and its susceptibility could be even lower than that of a conventional vessel. Likewise, vulnerability is also affected by operational aspects, such as watertight doors left open at the moment of impact of a weapon or poor performance of the fire-fighting parties, but these should be very unlikely events onboard a naval ship. Therefore the intrinsic design part of a naval vessel is almost decisive for the probability of survival after a weapon impact.

The tendency during recent decades in surface naval ship design was to assess and minimize susceptibility through detailed signature management. Therefore the probability of detection was usually estimated and it was considered as input in scenarios simulations. On the other hand the probability of staying afloat and upright was less frequently taken into account. Most of the simulations assumed a single-hit-kill probability equal to 1.0 for small naval ships whereas 2 hits were considered adequate for the sinking of larger vessels. Thus the defence analysis was actually never treating the vulnerability as a probability. For the naval architect it is usually enough to assess the adequacy of its design with respect to vulnerability through the use of the damaged stability requirements introduced by the various navies, such as those used by the USN and the UK MoD, depicted in Table I (*Surko, 1994*).

Table I: Current UK & US Damage Stability Criteria for surface warships

CRITERIA	UK NES 109		U.S.N. DDS-079	
DAMAGE LENGTH	$L_{WL} < 30m$	1-compartment	$L_{WL} < 100ft$	1-compartment
	$30m < L_{WL} < 92m$	2 comp of at least 6m	$100ft < L_{WL} < 300ft$	2 comp, at least 6m
	$92m < L_{WL}$	$\max\{15\% L_{WL} \text{ or } 21m\}$	$300ft < L_{WL}$	$15\% L_{WL}$
PERMEABILITY	Watertight Void	97 %	Watertight Void	95 %
	Accommodation	95 %	Accommodation	95 %
	Machinery	85 %	Machinery	85 % - 95 %
	Stores etc	60 %	Stores etc	60 % - 95 %
Angle of list or loll	$< 20^\circ$		List $< 15^\circ$	
GZ at C (Fig.5)	60 % of GZmax			
Area A1	$> 1.4 \text{ Area A2}$		$> 1.4 \text{ Area A2}$	
Longitudinal GM	> 0		-	
Buoyancy	Longitudinal trim Less than that required to cause downflooding		3 in margin line	

The difference between the susceptibility and vulnerability is that the first can be altered even in later design stages, even during the operational life of the ship (use of Radar Absorb Materials-RAM, infrared signature suppression devices and low emission paints), whereas most of the issues that affect vulnerability will almost certainly characterise the ship throughout her life. Therefore a generic naval ship design methodology for enhanced survivability should consider the minimization of the ship's vulnerability in the early concept design phase.

3. Vulnerability estimation

The authors have presented earlier a generic concept for the design of both merchant and naval ships of enhanced survivability, *Papanikolaou and Boulougouris (2000)*. It is based on the fundamental probabilistic damage stability concept originally introduced by *Wendel (1960)* and its derivatives -in particular *IMO Resolution A.265* for passenger ships and *IMO MSC.19 (58)* for cargo ships- for the assessment of ship's survival capability after damage. It recognizes the following probabilities of events relevant to the ship's damage stability:

- The probability that a ship compartment or group of compartments *i* may be flooded (damaged), p_i .
- The probability of survival after flooding the ship compartment or group of compartments *i* under consideration, s_i .

The total probability of survival is expressed by the attained subdivision index *A* which is given by the sum of the products of p_i , and s_i for each compartment and compartment group, *i*, along the ship's length:

$$A = \sum_i p_i \cdot s_i \quad (2)$$

The IMO regulations require that this attained subdivision index should be greater than a required subdivision index *R*, which for passenger ships is presently determined by ship's length, the number of persons and the extent of life-saving equipment onboard. This value is a measure for the ship's probability of surviving a random damage event and it is obvious that this value should increase with the number of persons onboard the ship. The factors in the formula determining *R* are so selected to correspond to the mean values of the attained subdivision indexes of a sample of ships with acceptable survival characteristics.

Likewise for naval ships there is a probability distribution of a hit by a threat weapon at any position along the ship. This depends on both the characteristics of the threat (weapon) and the characteristics of the target (ship). It is easily understood that this distribution relates *susceptibility* with *vulnerability* characteristics. During the initial stages where there is a lack of actual estimations for the signature distribution along the ship we may assume that the probability of weapon impact along the ship follows a basic mathematical distribution, such as the piecewise linear or the normal one.

We may assume as a working hypothesis for air-to-surface missile (ASM) threats a piecewise linear distribution with the maximum probability amidships, whereas for contact mines we may assume a linear distribution (*Harmsen and Krikke, 2000*) (see Fig.1).

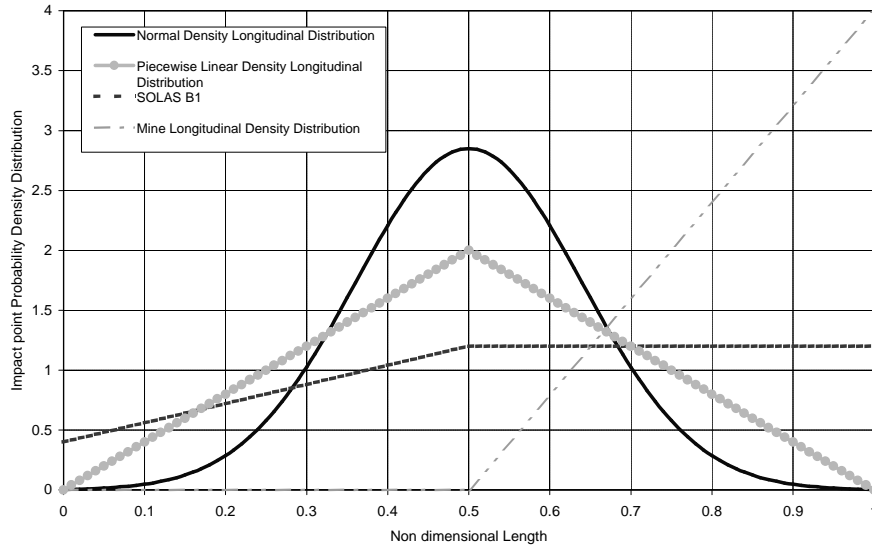


Fig.1: Comparison of alternative longitudinal damage distributions, *Boulougouris (2003)*

Thus the impact point probability density function in the missile's case with a piecewise linear distribution is:

$$\text{Imp}(x) = \begin{cases} 4x, & x \leq 0.5 \\ -4x + 4, & x > 0.5 \end{cases} \quad (3)$$

whereas in the case of a normal distribution it would be:

$$\text{Imp}(x) = \frac{1}{\sqrt{2 \cdot p \cdot s}} \cdot \exp \left[-\frac{1}{2s^2} \cdot (x - 0.5)^2 \right] \quad (4)$$

In Fig.1 both these distributions are compared with the longitudinal distribution assumed in SOLAS A.265 for passenger ships.

Based on the concept of the *Damage Function* used in the theory of Defence Analysis, the fraction of the target assumed to be damaged within a radius r from the impact point is assumed to follow the log-normal distribution given by the following formula, *Przemieniecki (1994)*:

$$d(r) = 1 - \int_0^r \frac{1}{\sqrt{2pb} \cdot r} \exp \left[-\frac{\ln^2(r/a)}{2b^2} \right] dr \quad (5)$$

where:

$$a = \sqrt{R_{SS} R_{SK}}, \quad b = \frac{1}{2\sqrt{2}z_{SS}} \ln \left(\frac{R_{SS}}{R_{SK}} \right),$$

R_{SK} is the sure kill radius which means that $d(R_{SK}) = 0.98$

R_{SS} is the sure save radius which means $d(R_{SS}) = 0.02$

z_{SS} constant equal to 1.45222

Based on the above the damage length probability density distribution is given by the formula:

$$\text{Dam}(y) = \frac{1}{\sqrt{2pb} \cdot y} \exp \left[-\frac{\ln^2(y/a)}{2b^2} \right] \quad (6)$$

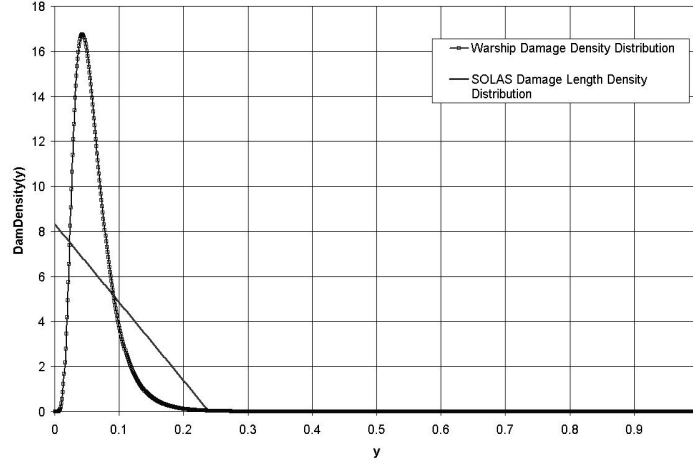


Fig.2: Damage Density Distribution, *Boulougouris (2003)*

The difference between this distribution function for naval ships and the one used in the SOLAS damage stability probabilistic regulations Fig.2 should be noted.

The damage extent ranges of naval ships may result from test analysis, analysis of data from actual engagements, empirical formulas linking the damage range with the type and the weight of the warhead or from the use of damage lengths defined in current deterministic damage stability regulations for naval ships.

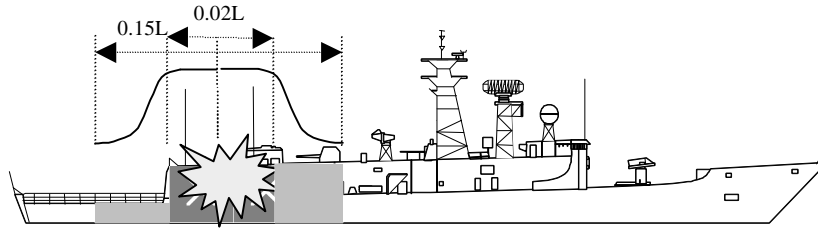


Fig.3: Damage extent on naval ship profile

In the later case, which is the one followed by the authors, a first approximation of the R_{SS} can be taken according to NES 109 and DDS-079 and it would be $0.15L$ (see Fig.3). The R_{SK} has been assumed equal to $0.02L$.

Based on the above the probability of a damage lying between the boundaries x_1 and x_2 of a naval ship's compartment is:

$$p_i|_{x_1}^{x_2} = \int_0^y Dam(y) \int_{x_1 + \frac{y}{2}}^{x_2 - \frac{y}{2}} Imp(x) dx dy \quad (7)$$

Substituting the piecewise linear $Imp(x)$ into equation (7) we obtain after some algebraic transformations:

If $x_1 < 0.5$ and $x_2 < 0.5$

$$\int_0^y Dam(y) \int_{x_1 + \frac{y}{2}}^{x_2 - \frac{y}{2}} 4x dx dy =$$

$$(x_2^2 - x_1^2) \left[\operatorname{erf} \left(\frac{\ln \left(\frac{y}{a} \right)}{\sqrt{2}b} \right) + 1 \right] - (x_2 + x_1) a \left[\operatorname{erf} \left(\frac{\ln \left(\frac{y}{a} \right)}{\sqrt{2} \cdot b} - \frac{1}{\sqrt{2}} b \right) \exp \left(\frac{b^2}{2} \right) + \exp \left(\frac{b^2}{2} \right) \right] \quad (8)$$

If $x_1 > 0.5$ and $x_2 > 0.5$

$$\begin{aligned} & \int_0^y \operatorname{Dam}(y) \int_{x_1 + \frac{y}{2}}^{x_2 - \frac{y}{2}} (-4x + 4) dx dy = \\ & = (x_2 + x_1 - 2) a \left[\operatorname{erf} \left(\frac{\ln \left(\frac{y}{a} \right)}{\sqrt{2} \cdot b} - \frac{1}{\sqrt{2}} b \right) \exp \left(\frac{b^2}{2} \right) + \exp \left(\frac{b^2}{2} \right) \right] - \\ & - (x_2^2 - x_1^2 - 2x_2 + 2x_1) \left[\operatorname{erf} \left(\frac{\ln \left(\frac{y}{a} \right)}{\sqrt{2}b} \right) + 1 \right] \end{aligned} \quad (9)$$

If $x_1 < 0.5$ and $x_2 > 0.5$

$$\begin{aligned} & \int_0^y \operatorname{Dam}(y) \left[\int_{x_1 + \frac{y}{2}}^{\frac{1}{2}} 4x dx + \int_{\frac{1}{2}}^{x_2 - \frac{y}{2}} (-4x + 4) dx \right] dy = \\ & = -a^2 \sqrt{2}b \cdot \left(\frac{1}{2} \operatorname{erf} \left(\frac{\ln \left(\frac{y}{a} \right)}{\sqrt{2}b} - \sqrt{2}b \right) \sqrt{p} \cdot \exp(2b^2) + \frac{1}{2} \sqrt{p} \cdot \exp(2b^2) \right) - \\ & - (x_1 - x_2 + 1) \cdot a \left[\operatorname{erf} \left(\frac{\ln \left(\frac{y}{a} \right)}{\sqrt{2} \cdot b} - \frac{1}{\sqrt{2}} b \right) \exp \left(\frac{b^2}{2} \right) + \exp \left(\frac{b^2}{2} \right) \right] - \\ & - \left(x_1^2 + x_2^2 - 2x_2 + \frac{1}{2} \right) \cdot \left[\operatorname{erf} \left(\frac{\ln \left(\frac{y}{a} \right)}{\sqrt{2}b} \right) + 1 \right] \end{aligned} \quad (10)$$

Using the above formulas (8), (9) and (10) we may calculate the probability of a hit occurring at any compartment along a naval ship. Additionally we may also calculate the probability that the ship may lose one of its capabilities such as propulsion, due to the flooding of a combination of compartments. If j is the combination of compartments that includes the main machinery compartments or other vital equipment required by a ship's function (e.g. mobility), then the probability that the ship will remain to a certain degree operable after damage Sf , is:

$$Sf = \sum_i p_i \cdot s_i - \sum_j p_j \cdot s_j \quad (11)$$

The vertical extent of damage may also vary depending on the weapon's characteristics. In a surface combatant such as a frigate or a destroyer there are 3 vertical watertight boundaries as shown in Fig.4, namely the tanktop, the damage control deck and the main deck.

An Air-to-Surface Missile (ASM) is more likely to cause larger damage above waterline, leaving the tanktop deck probably intact, whereas an underwater weapon such as a contact-mine or a torpedo is likely to leave the damage control deck intact. A linear distribution for the probability density function can be used in this case with the maximum occurring at the main deck depth and the minimum at the keel in case of a hit by an ASM and vice-versa in case of an underwater weapon (see Fig.4). In order to take into account both threats a weighting factor can be applied according to an operational analysis of the potential threats. The damage penetration distribution is not an ‘issue’ for surface combatants, as commonly a longitudinal watertight subdivision that may result to asymmetrical flooding conditions is avoided by design.

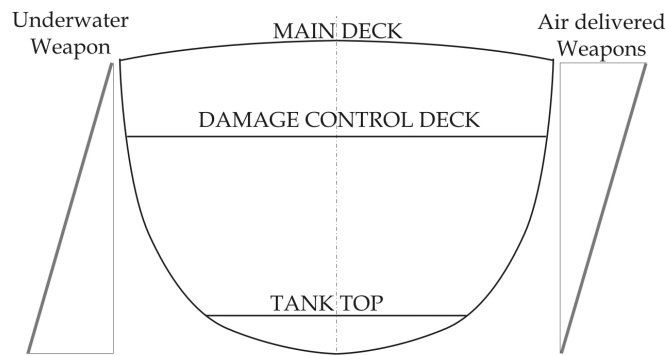


Fig.4: Vertical Watertight Boundaries

4. Assessment of Survivability

The probability of survival of a ship after damage in waves can be estimated using:

- A time domain nonlinear ship motions - capsizing simulation code (*parametric deterministic study approach*).
- A probabilistic damage stability quasi-static approach adjusted for the currently valid deterministic criteria for naval ships (*probabilistic study approach*).

4.1. Time domain capsizing simulation approach

In order to calculate the probability of ship's survival in damage condition and for a variety of environmental and ship loading conditions the most advanced methodology would be to use a direct time domain ship motions simulation program, such as *CAPSIM*, Papanikolaou *et al.* (2000) and Spanos (2002), *FREDYN*, de Kat and Peters (2002), or *PROTEUS*, Jasionowski (2002), enabling the prediction of survival after damage with satisfactory accuracy. Related work on the assessment of naval ships survival by time domain simulations has been presented in several recent publications, Alman *et al.* (1999), Harmsen and Krikke (2000), de Kat and Peters (2002). Herein, the ship's survivability is not addressed on the basis of hydrostatic properties, like the GZ-curve, but by analysis of the dynamic behaviour of the damaged ship in waves, considering the in- and outflow of water from/to damaged compartments. The time domain simulation enables the determination of the combinations of exciting wave height and period the ship will survive, capsize or sink, assuming the wave scatter in the area of ship operation. The survival probability after damage is the summation of the probability of occurrence of all wave height-wave period combinations for which the ship survives.

The drawback of this approach is the vast number of required calculations and the associated computing time that makes it difficult to implement it in a formalised optimisation procedure, as proposed herein. For a typical frigate with 12 compartments, assuming 6 flooding combinations, 5 damage causes, 4 hole sizes, 4 ship speeds, 8 wave headings and 2 scenarios for the consideration of extinguishing water, the necessary number of simulation runs for the evaluation of just one design is

92160 and the associated computing time on a desktop computer several tens of hours, *Harmsen and Krikke (2000)*.

4.2. Probabilistic damage stability quasi-static approach

A more efficient methodology to implement the suggested survivability assessment procedure within a ship design optimisation scheme is an approach that considers the probability of survival based on quasi-static survival criteria, like those of the RN and the USN. They take into account data of real damage incidences of WWII and they have proved to be reliable until today, in so far they appear satisfactory to not have been changed over a long period of time.

The philosophy for transforming these deterministic criteria into a set of rational probabilistic approach criteria will be herein based on an approach similar to the *IMO Resolution A.265* for passenger ships. It is well established that in all relevant criteria there is an underlying assumption that the sea conditions at the time of damage are "moderate". This constraint could be lifted if there was a requirement for specific survival sea state in case of damage. This would allow the correction of these requirements by consideration of the probability of exceedence of the wave height considered as basis for the current deterministic RN and USN criteria, namely a significant wave height, H_S , of 8 ft. This wave height was used in the above criteria for the determination of θ_{roll} , namely the roll amplitude due to wave action. It was also the underlying assumption behind the guidelines for establishing the watertight features/closures to prevent progressive flooding. Thus, any attempt to change the wave amplitude must take into account changes in both θ_{roll} as well as the margin line or equivalent. Another important environmental parameter is the wind speed. Given the small probability of exceeding the values given by the U.S. Navy standards for warships (namely, about 33 knots for a 3500 tons frigate), this value could be left unchanged. As a working hypothesis it is proposed that the following *survival criteria* be applied in the frame of a probabilistic approach to the survivability of *naval ships* (see Fig.5 for the meaning of the various notions of the righting arm curve). For intermediate stages, interpolant values can be used.

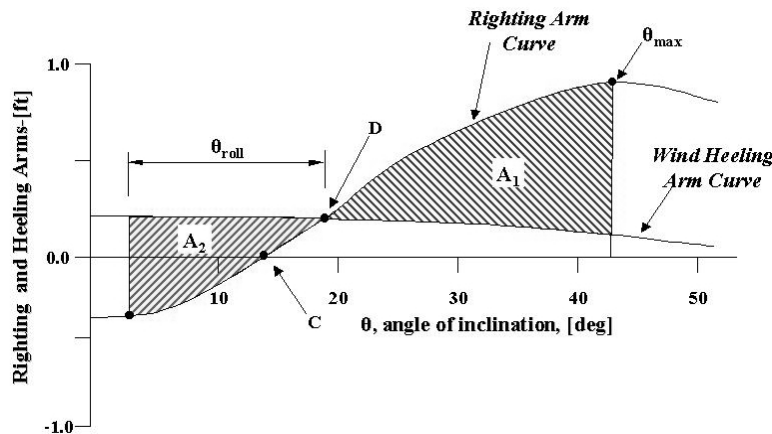


Fig.5: Survival Criteria

Table II: Proposed Damage Stability Criteria for Warships

$S_i = 1$	$\theta_{roll} = 25 \text{ deg}$ Wind speed = accord. to DDS-079-1. $A_1 \geq 1.4 A_2$ Min Freeboard $\geq 3in + 0.5 \times (H_S(0.99) - 8 \text{ ft})$
$S_i = P(H_S \leq 8 \text{ ft})$	Ship meets DDS-079 damaged stability criteria
$S_i = 0$	$\theta_{roll} = 10 \text{ deg}$ Wind speed $\leq 11 \text{ knots}$ $A_1 \leq 1.05 A_2$ Margin line immerses.

For the probability distributions of wave exceedence in the probable area of operation we assumed $P(H_S \leq 8 \text{ ft}) = 0.60$ for the North Atlantic and $P(H_S \leq 8 \text{ ft}) = 0.90$ for the East Mediterranean Sea, *Athanassoulis and Skarsoulis (1992)*. Thus, a combatant, meeting the U.S. Navy criteria for warships, should have, according to the above-proposed criteria, a 60% probability of survival for a damage

length according to the existing regulations in the North Atlantic and a 90% probability of survival in the Mediterranean Sea. Obviously a similar methodology can be introduced for auxiliary naval ships.

5. Naval ship design optimisation model

5.1. Optimization procedure

Recent advances in the optimal design techniques and increased computing power allowed the introduction of a wide range of tools for the exploration of the design space once it is described in a parametric way. Given the fact that there was very little information about the mathematical behaviour of the objective space of the present problem –especially with respect to the survivability index for which there appears that it is of multi-modal type– and there were multiple conflicting objectives and constraints, the adoption of multi-objective GAs appeared like the only solution to the set optimization problem, *Sen and Yang (1998)*.

The GAs were herein implemented by use of the general-purpose optimization software package *modeFRONTIER* (see *ES.TE.CO, 2003*), based on the positive experience with similar applications to the optimisation of Ro-Ro Passenger ships, *Boulougouris et al. (2003)*. The interfaced *NAPA-FRONTIER* optimisation procedure, as implemented in the present naval ship optimisation study, is shown in Fig.6. A parametric ship model was created in the environment of the well-known ship design software package *NAPA*, *Napa Oy (2001)* by use of *NAPA Basic* programming language. Assuming that the hull form and the main layout concept were developed independently at the previous design stages, the vessel's watertight subdivision is parametrically generated.

For each design variant the survivability index, the probability of mobility failure, the transverse bulkhead area as a measure to minimise structural weight, the length of the shaft lines and the length of each engine room (main or auxiliary) are calculated. The main features of the employed procedure are outlined in the following paragraphs.

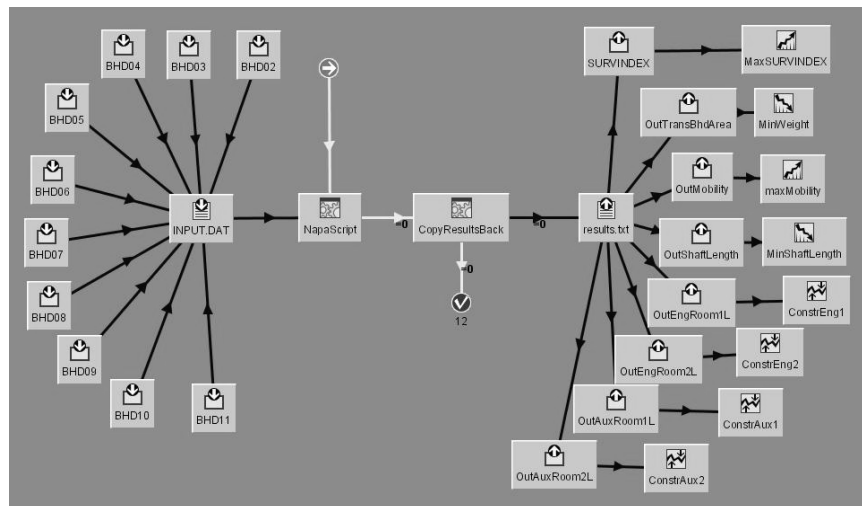


Fig.6: Frontier Optimization Procedure

5.2. Parametric model

A parametric geometry model for ship's compartmentation has been developed to enable the implementation of proposed methodology within a formal optimisation procedure. This has been achieved by use of the *NAPA* design software platform. Several *NAPA* MACROS were developed enabling the communication of parametrically generated geometry data with the *NAPA* design database and relevant *NAPA* calculation modules. The hullform (see Fig.7) used at the present version of the model is preserved in all design alternatives and the internal compartmentation (see Fig.8) is automatically generated by use of a set of design variables. These include: the number of transverse

bulkheads, their frame location, the height of the double bottom and the compartments where the main and the auxiliary engine rooms are located.

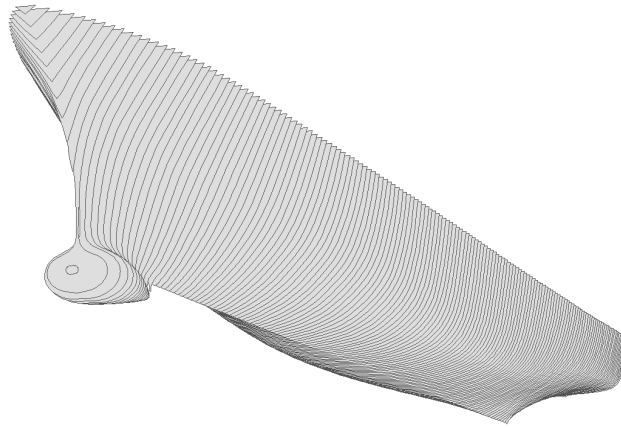


Fig.7: Hull form of sample naval ship

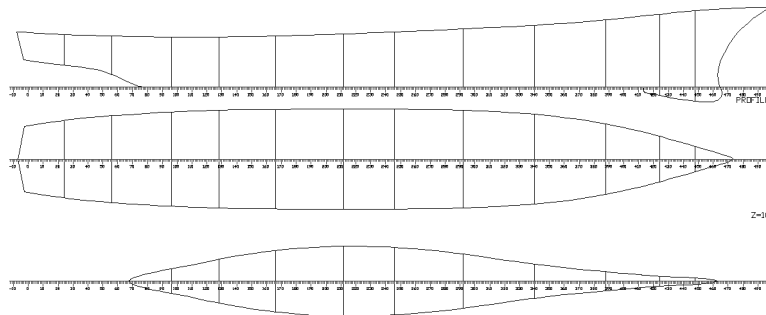


Fig.8: Initial Compartmentation

The model initialises its configuration reading an external input file where the user specifies the ship's intact condition in terms of draught, trim and the metacentric height (GM). The user may also define the frame spacing for positioning the transverse bulkheads. The input file includes also:

- The probability of non-exceedance of a significant wave height of 8 ft in the area of operation.
- The wave height with a 99% probability of non-exceedance.
- The sure save length of damage (RSS) that is subsequently used in the probabilistic calculations.

5.3. Objective functions

The designer has to take into account during the initial stages of the design process not only the impact of the watertight compartmentation on the probability of survival in case of damage, but also on the structural weight, the arrangement of the machinery spaces, the topside layout (which affects its signatures) and finally to the functionality of the ship. Thus the objective functions that have to be optimised and the constraints to be taken into account in this design model could be:

- The maximisation of the Survivability Index
- The maximization of the probability that the ship will maintain her mobility
- The minimisation of the transverse bulkhead area as a measure to minimise structural weight
- The minimisation of the length of the shaft lines
- The satisfaction of the minimum engine room requirements posed by the requested space for the gas turbines and their auxiliary equipment.

5.4. Genetic Algorithms

The presence of multiple and conflicting objectives, the large and complex solution space and the complex characteristics of the objective functions (especially those of the survivability index), favour

the use of a stochastic optimization process such as the Genetic Algorithms (GA) (see *Goldberg, 1989*).

The *modeFRONTIER* implements a Multiple Objectives Genetic Algorithm (MOGA) optimization scheduler (see *E.STE.CO, 2003*) that searches for the Pareto-optimum solutions. The drawback of the procedure is the large number of direct calculations required to converge to optimum solutions. The advantages of the algorithm are that:

- It avoids converging into local optimum solutions in the design space
- Starting from the initial population, it allows it to evolve in such a way that some individuals can meet different objectives. This results to a “set of best designs” (*Pareto set*)
- It poses no limitation to the characteristics of the objective function

A multi-objective problem may be treated with three different strategies using a GA according to *Sen and Yang (1998)*:

- Make the multiple criteria decisions first and arrive to a composite measure of fitness by combining the different criteria, and then use the composite measure to search for the best solution
- Conduct the search to assemble a range of possible solutions and then select one or more of these on the basis of multiple criteria decision making
- Combine the search with the Multiple Criteria Decision Making (MCDM)

The third solution is herein implemented within the *modeFRONTIER*. The basic pattern is as follows:

- A multiple objective search is performed to obtain an approximate idea of the Pareto surface.
- Multiple criteria choice or ranking is applied to capture the preferences of the decision maker.

The MOGA optimization scheduler has the following parameters:

- The initial population. An initial population of 42 designs was used.
- The number of generation.
- The probability of directional cross-over that is a proprietary operator that gives efficiency to the algorithm but decreases its robustness. A value of 0.75 was used.
- The probability of selection, which gives the probability that design configurations are not changed during the evolution. A value of 0.05 was used.
- The probability of mutation that gives the probability that a design configuration is randomly perturbed. A value of 0.01 was used.
- The DNA String Mutation Ratio that gives the percentage of the individual DNA that is perturbed by the mutation operator. A value of 0.05 was used.
- The usage (or not) of Elitism, which ensures that the best solutions (for each objective) are preserved during the evolution
- The MOGA type. There are three types:
 - o MOGA - Generational Evolution that works on a set of design configurations that are periodically updated when one generation is completed
 - o MOGA - Steady Evolution that uses all the computed configurations as soon as they are available in a first in - first out mode.
 - o MOGA - Adaptive Evolution, where the choice of the genetic algorithm operators is done dynamically during the search

5.5. Sample Case study

A standard destroyer hullform, *Brown and Deybach (1998)*, having an intact draught of 6.34m and GM of 1.26m, was optimised using the proposed method. The case study was performed for a hypothetical operational sea environment having 90% probability of non-exceedence of the 8ft wave height and a 99% probability of not exceeding a significant wave height of 3 meters. Figs.9 to 13 show a typical sample of results, encompassing the assessment of 924 generated designs. The resulting survivability index values correspond well to the values calculated for a single frigate design by *Harmsen and Krikke (2000)* for an ASM threat, using the FREDYN code.

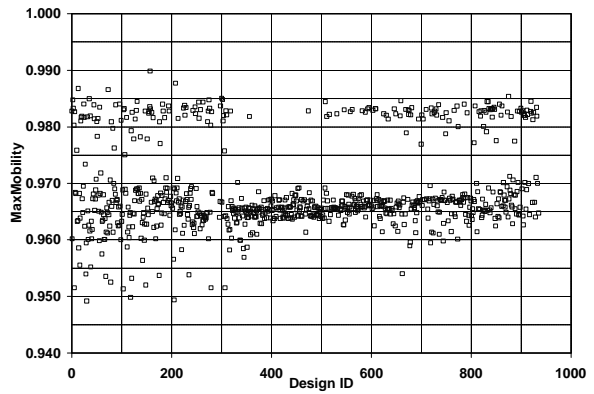
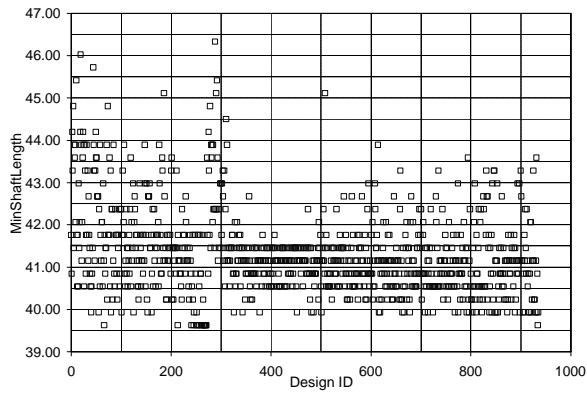
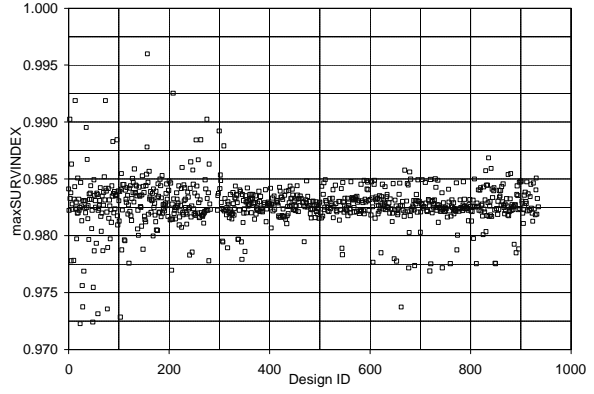
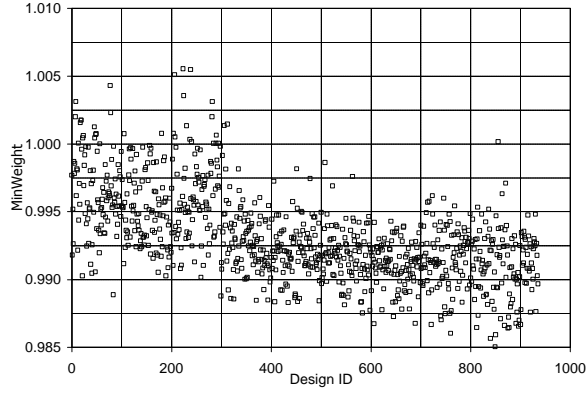


Fig.9: History Charts of Objective Functions

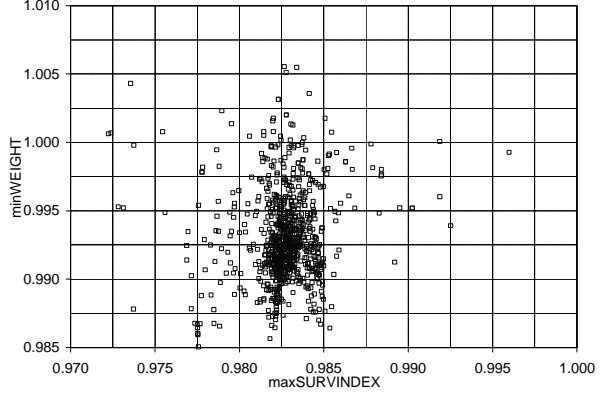
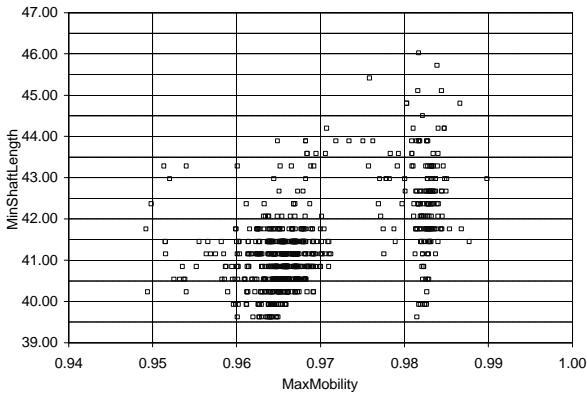


Fig.10: Scatter diagram of Mobility vs. Shaft Line Length

Fig.11: Scatter diagram of Survability Index vs. Weight

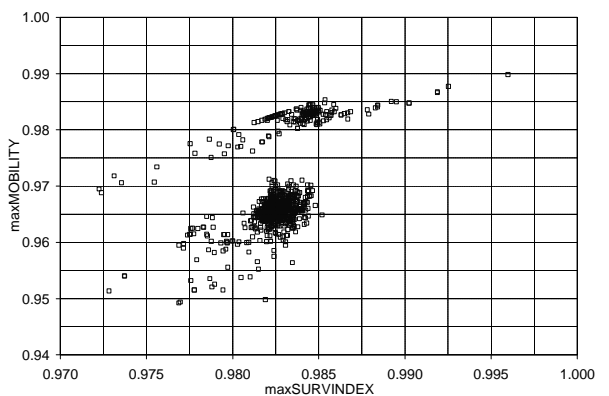
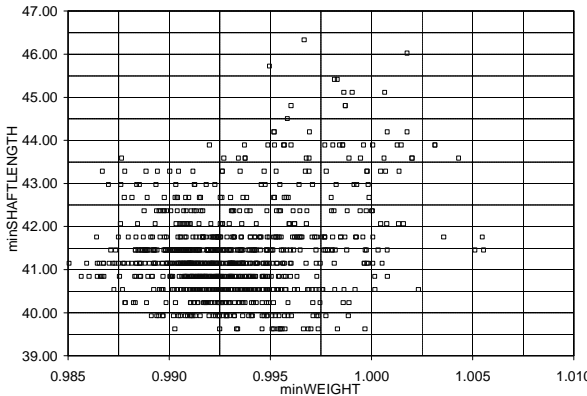


Fig.12: Scatter diagram of Weight vs. Shaft Line Length

Fig.13: Scatter diagram of Survability Index vs. Mobility

Each MOGA loop requires approximately 1.5 min time on a PC using a Pentium 4 CPU at 2.4GHz with 512MB RAM. It is obvious that even if this initial ship design procedure is very efficient for the assumed sea environment and damage extent characteristics, that some improvements are still possible.

5.6. Selection of the optimum design

Using the initial mapping of the design space performed by MOGA, the optimum design can be selected using an appropriate multi-criteria decision method such as the additive utility functions. The decision maker may compare pairwise a collection of non-dominated solutions obtained from the MOGA's search or direct specify the relative importance between design's attributes. The information on these pairwise comparisons result to one of two kinds of ordering:

- Solution i is preferred to Solution j
- Solution i is considered to be just as attractive as Solution j

Using these relations a set of utility functions for each criterion can be derived analytically for all the attributes of a design. The utility functions are then combined, forming a composite fitness function that captures the preference structure of the decision maker. Using this fitness function all the designs can be ranked. In Fig.14 a sample ranking is shown, using direct assignment of equal importance to all objectives. Table III contains the attribute values of the optimum design.

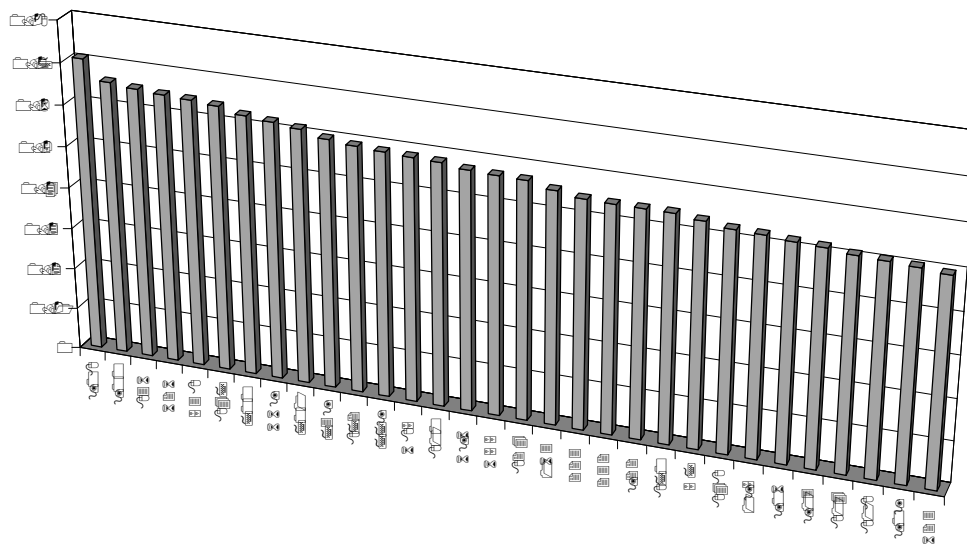


Fig.14: Scatter Diagram of Survivability Index vs. Weight

Table III: Objective values for the optimum design using MCDM

Design No.	ShaftLength	Weight	Mobility	SurvIndex	Rank Value
908	41.7576	0.98641	0.98536	0.98536	0.708306

6. Conclusions

A methodology for the optimisation of the design of naval ships for enhanced survivability has been presented. The method has been applied to the optimisation of a standard destroyer using a developed interface of the naval architectural design software platform *NAPA* and the optimisation package *modeFRONTIER*.

Further developments of this model include consideration of more design parameters, of relevance to the design of surface combatants such as:

- The volumetric constraints imposed by the weapon systems.
- The topside design and signature characteristics.
- The functionality optimisation of the internal arrangement.
- The impact on survivability of intermediate stages of flooding.

This will enhance the capabilities of the model allowing the naval ship designer a better control and understanding of the compromises on the way to optimal designs. Finally, an exploration and systematic study of the variety of parameters affecting a naval ship's survivability might eventually lead to the formulation of Required Subdivision Indices for naval ships, as a function of ship type, ship size and operational profile.

References

- ALMAN, P.R.; MINNICK, P.V.; SHEINBERG, R.; THOMAS, W.L. (1999), *Dynamic capsize vulnerability: reducing the hidden operational risk*, SNAME Annual Conference, Vol. 107
- ATHANASSOULIS, G.; SKARSOULIS, M. (1992), *Wind and wave atlas of the north-eastern Mediterranean sea*, NTUA-SMHL Publications
- BALL, R.E.; CALVANO, C.N. (1994), *Establishing the fundamentals of a surface ship survivability design discipline*, Naval Engineers J. 106/1, pp.71-74
- BOULOUGOURIS, E.; ZARAPHONITIS, G.; PAPANIKOLAOU, A. (2003), *Optimization procedure for fast ferry watertight subdivision (WP 5)*, ROROPROB Project Report, Project funded by the European Commission, contract G3RD-CT-2000-0030, NTUA-SDL
- BOULOUGOURIS, E.K. (2003), *Ship design optimization for enhanced survivability after damage for Ro-Ro passenger and naval ships*, Dr.-Eng. Thesis (in Greek), National Technical University of Athens
- BROWN, A.J.; THOMAS, M. (1998), *Reengineering the naval ship design process*, From Research to Reality in Ship Systems Engineering Symp., University of Essex, ASNE, pp.277-289
- BROWN, A.J.; DEYBACH, F. (1998), *Towards a rational intact stability criteria for naval ships*, Naval Engineers J. 110/1, pp.65-77
- De KAT, J.O.; PETERS, A.J. (2002), *Model experiments and simulations of a damaged frigate*, Proc. X congress of the Int. maritime Association of Mediterranean, IMAM 2002, Crete, May 2002
- ES.TE.CO., *modeFRONTIER*, Version 2.5.0, <http://www.esteco.it/>, Trieste, Italy
- GOLDBERG, D. (1989), *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley Publ.
- HARSEN, E.; KRIKKE, M. (2000), *A probabilistic damage stability calculation method for naval vessels*, 7th Int. Conf. Stability of Ships and Ocean Vehicles, STAB2000, Tasmania, pp. 330-350
- IMO RESOLUTION A.265 (VIII) (1973), *Regulations on subdivision and stability of passenger ships as an equivalent to Part B of Chapter II of the Int. Convention for the Safety of Life at Sea, 1960*, IMO 1973
- IMO RESOLUTION MSC.19 (58) (1990), *New regulations for subdivision and damage stability for dry cargo ships built on or after 920201*, 25 May

JIASIONOWSKI, A. (2002), *An integrated approach to limit state performance assessment*, PhD thesis, University of Strathclyde, Glasgow

NAPA Oy (2001), *NAPA Release 2001.2*, <http://www.napa.fi/>, Helsinki, Finland

PAPANIKOLAOU, A.; BOULOUGOURIS, E. (2000), *On a rational approach to the assessment of survivability of surface naval and merchant ships*, 9th Congr. Int. Maritime Association of Mediterranean, IMAM 2000, Ischia

PAPANIKOLAOU, A.; ZARAPHONITIS, G.; SPANOS, D.; BOULOUGOURIS, E.; ELIOPOULOU, E. (2000), *Investigation into the capsizing of damaged ro-ro passenger ships in waves*, 7th Int. Conf. Stability of Ships & Ocean Vehicles, STAB2000, Tasmania

PRZEMIENIECKI, J.S. (1994), *Mathematical methods in defense analyses*, 2nd Edition, American Institute of Aeronautics and Astronautics

REESE, R.; CALVANO, C.; HOPKINS, T. (1998), *Operationally oriented vulnerability requirements in the ship design process*, Naval Eng. J. 110/1, pp.19-34

SAID, M.O. (1995), *Theory and practice of total ship survivability for ship design*, Naval Eng. J. 107/3, pp.191-203

SEN, P.; YANG J-B. (1998), *Multiple criteria decision support in engineering design*, Springer

SPANOS, D. (2002), *Numerical simulation of flooded ship motions in seaways and investigation of the behaviour of passenger/ro-ro ferries*, Doctoral thesis, National Technical University of Athens

SURKO, S.W. (1994), *An assessment of current warship damaged stability criteria*, Naval Engineers Journal, Vol.106, No.2, pp. 120-131

VASSALOS, D. (1999), *Shaping ship safety: The face of the future*, Marine Technology 36/2, pp.61-73

WENDEL, K. (1960), *Die Wahrscheinlichkeit des Überstehens von Verletzungen*, Ship Technology Research 7, pp.47-61

Knowledge and Data Reuse in Ship System Design and Engineering

Jan Jaap Nieuwenhuis, Schelde Naval Shipbuilding, jan-jaap.nieuwenhuis@schelde.com
Ubald Nienhuis, Delft University of Technology, u.nienhuis@wbmt.tudelft.nl

Abstract

This paper gives an overview of the possibilities and consequences of reusing knowledge and data in ship system design. A number of reuse scenarios have been worked out and the consequences are evaluated.

1. Introduction

Most modern day ships are developed as engineered-to-order, one-off products, requiring a substantial design, engineering and production effort. Pushed by fierce competition shipbuilders continuously have to find ways to reduce their product development time and costs. Many attempts have been made to achieve this goal; advanced computer systems have been introduced at many yards and concepts like 'Concurrent Engineering', 'Total Systems Approach', 'Integrated Product and Development principles' have been implemented during the past decades, *Tan et al. (1998)*. It is clear that these attempts have improved the efficiency of the ship development process, however there still are possibilities for further improvements. The option considered in this project is to increase and systematize knowledge and data reuse in 'engineering'¹.

Research projects in other design engineering industries have shown, that engineers spend a lot of time looking for information, documenting, communicating or negotiating, sometimes even more than 50% as among others is shown by *Bucciarelli (1984)*. Other researchers showed, that over 50% of the work done by a designer on a day-to-day basis consists of routine design, based on past design solutions, *Moore (1993)*. Therefore systematically reusing design information offers a good chance to strongly reduce product development time, in some occasions even up to more than 50% as shown by *Baya (1995)*. Although ship design differs from other engineering design disciplines, the similarities are sufficient to expect a major decrease in ship development time and costs too.

In this project, a method or tool will be developed to support a shipyard in making a well-founded decision between different reuse possibilities. This is done based on an evaluation of the consequences and feasibility of reuse for the different parts of the ship development process. Along the way, a small number of knowledge and data reuse scenarios will be implemented at limited scale in practice, to materialize the concept and to come up with reliable measurements of quantifiable consequences. In this paper an overview of the possibilities and consequences of knowledge and data reuse in ship system design is given and two of the possible scenarios to implement reuse are evaluated.

2. Knowledge and data in ship system design

Ship system design has an important place in the overall design process. During system design, a significant part of the total engineering hours² are spent (about 15% for complex vessels). Furthermore, a major part of the costs are fixed during system design, as most of the decisions what to buy and where to buy are made in this stage of the ship design process³.

¹ All design activities between contract design and delivery, like system design, detailed arrangement design, detailed structural design and production design.

² Engineering can take up to ten thousands of hours for relative simple ships, hundred thousands of hours for navy ships like frigates and several million hours for ships like aircraft carriers;

³ Considering the fact that about 70% of a ship's acquisition costs consist of bought in labour and material, it is clear that the decisions what and where to buy can have a considerable influence on the total costs, *Aalbers (2003)*.

Design problems are typically characterised by a large amount of very different kinds of knowledge, which all have to interact in a well defined and tuned manner, *Cutkosky et al. (1993)*. Ship system design does not form an exception. Most ship system design work can be classified as 'routine design'. In routine design, the attributes defining the system and the strategies and methods to attain them are usually known. To find out what kind of knowledge and data⁴ is available for reuse in ship system design, the design process of a ship's Black Water System (BWS) is analysed. 'Black water' is water coming from toilets and medical facilities and contains bacteria that can be harmful for the environment. In some areas, treatment of black water is thus required before discharge is allowed. Treatment can for example be done in a biological way (bacteria), with membranes, or with chemicals. A BWS takes care of the total process of collecting, storage and treatment of black water.

2.1 BWS design process

The process analysis is based on interviews with a number of experienced designers of 'Schelde Naval Shipbuilding', a well-known Dutch designer and builder of naval vessels. In short, the steps required to design and engineer a BWS are:

- analyze requirements, rules and regulations;
- collect all necessary relevant information (techniques to be used, calculation rules, component suppliers, etc);
- define system configuration if not prescribed by the client (treatment type, transfer system type, etc);
- calculate capacities for black water treatment plant, storage tanks and discharge pumps (this can either be done by the yard, the material supplier or both);
- request proposals for system components (most components are not manufactured by the yard, but bought in);
- evaluate proposals and choose the best one (usually the decision is made based on costs or dimensions/weight of the components);
- complete the BWS design (among other. calculate piping length, weight and costs);
- write system specification;
- define piping diameters and draw system diagram;

These steps make up 'ship system design'. Ship system design is followed by detailed arrangement design, wherein:

- detailed arrangement drawings of ship spaces are created (2D or 3D);
- pipes, cables and ducts are routed through the accommodation and engine room (usually in a 3D model of the ship's hull);
- arrangement drawings, piping/cabling/ducts and the ship's structural hull design are combined in one 3D product model;
- all necessary production drawings and production information (such as part lists) is derived from the 3D product model.

Although this is a simplified view of a ship's detailed design process (e.g. feedback or iteration steps are not shown) it does give a good reflection of the way most common ships and ship's systems are developed.

2.2 Knowledge and data used and generated during BWS design

Even during the design process of a relative simple system as a BWS, a wealth of knowledge and data is used and generated. Knowledge and data used and generated in ship system design are summarised in Table I. These different knowledge and data components interact in a very specific way in design problem solving. A design has to be generated in such a way that the requirements are fulfilled and that it is consistent with constraints in the domain knowledge (natural laws, technical laws, rules of behaviour, etc), *Klein (2000)*. This sounds quite simple, but as many people can tell from their own experience and as Klein shows this usually is very complicated in practice.

⁴ For definitions of data and knowledge, see among others *Thoben (1999)*.

Table I: Knowledge and data in ship system design

Knowledge/ data	Description	Example from BWS design and engineering	Explicit/Implicit ⁵ :
Domain knowledge	All general knowledge and data concerning a product design.	<ul style="list-style-type: none"> - Calculation rules; - physical laws to be obeyed; - knowledge about possible solutions; - documentation of previous projects. 	Explicit knowledge (documents, drawings, calculation models), as well as implicit knowledge (experience, skills)
Requirements, rules and regu- lations	All relevant requirements, rules and regulations a design has to obey or fulfill.	<ul style="list-style-type: none"> - Owner's requirements; - Regulations of class societies; - Legislation of national authorities; - Requirements coming from other parts of the design process (e.g. max. weight). 	Usually available in an explicit form.
Product data	All data directly related to the product under design.	<ul style="list-style-type: none"> - Results of calculation; - Drawings of the design; - Dimensions, weights; - System specifications. 	Explicit.
Design engi- neering rati- onale and intent	The 'how' and 'why' behind a design.	<ul style="list-style-type: none"> - Required steps to come to a successful design. - Reasoning behind choices; - Assumptions made for calculations; 	Only occasionally design engineering rationale is made explicit.

Besides the classification based on knowledge and data type, knowledge and data can also be classified according to their appearance form (explicit/implicit) and to the way it is captured or stored, adapted from *Kerssens-van Drongelen et al. (1996)*:

- brainware: knowledge and data in people's heads (experience, ideas);
- mediaware: text, drawings, photo's, movies, sound fragments etc.;
- physiware: prototypes, mock-ups, etc.;
- digiware: electronic files, virtual models, databases, etc.

In ship system design, all these appearance and storage forms can appear.

3. Reusing knowledge and data

Reuse of knowledge and data is in this project defined according to the definition as proposed by *Smith and Duffy (2001)*: " 're-use' can reflect the utilisation of any knowledge gained from a design activity and not just past design artefacts". Reusing different types of knowledge and data will however result in different consequences and benefits. For example, reuse of product data, often results in a regurgitation of the past, whereas reuse of domain knowledge and 'rationale and intent' results in an application of the past, *Duffy (1995)*.

The implementation of knowledge and data reuse in a design process can be done in a large number of ways. Differences can be made in 'what' is reused and 'how' reuse is carried out. An overview of 'what' kind of knowledge and data is available for reuse can be found in section 2.2. An overview of 'how' reuse can be implemented, is given by *Prieto-Diaz (1993)*:

- *By scope*:
 - o vertical: within a domain;
 - o horizontal: across several domains;
- *By mode*:

⁵ For definition of explicit/implicit see *Nonaka and Takeuchi (1995)*

- planned: systematic;
- ad-hoc: opportunistic;
- *By intention:*
 - black box: reuse as-is;
 - white-box: modifications possible before reuse;
- *By technique:*
 - generative: generating new products based on previous acquired knowledge and data;
 - compositional: composing new products out of knowledge and data developed for previous products.

In principle, all combinations of scope, mode, intention or technique can be applied in ship system design. In first instance however, only options considering vertical⁶, white-box⁷, generative or compositional reuse are investigated in this project.

3.1 Current practice of knowledge and data reuse

Of course there already is some reuse of knowledge and data in ship design in the current situation. In concept and preliminary design, the use of reference vessels as basis for a new design is widely applied. Furthermore a substantial part of the pre-design work is based on statistical data of previously designed ships (e.g.: weight estimation, resistance prediction). In engineering (detailed design), reuse of knowledge and data is done more ad-hoc. Some engineers use documentation of previous projects as an example or try to find information and data they can copy, but at most shipyards, this reuse process is not systematic and the extent and success mainly depends on the various employees. With respect to product data, most yards use a 'preferred part list', which provides for reuse of product data like bolts, valves bulkhead penetrations, etc. Some shipyards successfully offer (nearly) standardised ships (the ultimate degree of product data reuse), but in most shipbuilding sectors standardised ships do not sufficiently satisfy client requirements.

A number of tools and methods have been developed during the last few decades for reusing knowledge and data. Examples are:

- *expert systems:*
An expert system is a computer system that uses algorithms to capture domain-specific knowledge which enables the system to perform at the level of a human expert, *Park and Storch (2002)*. An expert system carries non-geometric data, which a CAD system could not support and guides a designer through the design process.

Within research projects expert systems have been developed for all ship design phases, however use in practice is still limited. Examples of expert systems are, *Park and Storch. (2002)*:

- Concept design:
 - HOSDES, a program from preliminary design, *Koops (1985)*;
 - SUBCEM, a knowledge based concept exploration model for submarine (concept) design, *Nat (1999)*;
 - DeSIS, a conceptual design tool for surface combatants, *Hees (2003)*
- Basic design:
 - ESMID for midship section design, *Lee et al. (1996)*
 - CICAD concurrent integrated CAD for surface ship design, *Tan and Bligh (1998)*
- Detail design:
 - SPIDA spatial intelligent design assistant for the spatial layout of catering decks, *Manfaat et al. (1998)*
 - Hull detailed design, *Bong et al. (1994)*

⁶ Most Dutch yards are specialized in a few ship types, not much need for horizontal reuse

⁷ Flexible reusable design solution are preferred, to increase possibilities for reuse and to facilitate the fulfillment of customer requirements.

- *parametric CAD*:
In parametric CAD systems like Solidworks, ProEngineer, or CADD5, a designer can give in (parametric) relations between different components. These relations can for example be based on calculation or production rules (e.g. relate pipe diameters to pipe length; when pipe lengths are changed, automatically pipe diameters change).
- *Knowledge-based engineering (KBE)*:
In KBE, knowledge and techniques used to design, analyse and manufacture a product are stored in a product model. Unlike CAD models, a KBE model not only contains geometric information, but all information required to design a product, resulting in a 'product model' [KTI, ICAD pages]. Other differences between KBE and parametric CAD are [KTI, ICAD pages]:
 - o parametric CAD does not automate the design, only adjustment of drawings is automated. In KBE a larger part of the design process can be automated;
 - o parametric CAD is geometry based and cannot easily handle other types of engineering knowledge;
 - o most parametric CAD systems produce geometric output, KBE systems can also produce different kinds of output, like reports or specifications;

KBE systems also show some resemblance with expert systems, however there are also some clear differences [KTI, ICAD pages]:

 - o most expert systems have not built in geometric primitives to describe mechanical products;
 - o expert systems usually use "if-then" rules, in KBE systems rules can take any format an engineer likes;
 - o in expert systems every rule needs to be checked after a change, which drastically decreases the efficiency. To resolve this, some expert systems offer the possibility to define 'meta-rules', however 'meta-rules' are hard to define and to debug. KBE-systems automatically keep track of the rules that need to be updated, and only updates those rules after a change.

KBE systems can be used in both generative, as well as compositional vertical reuse (if necessary coupled with a PDM system). KBE aims at reusing domain knowledge, rationale and 'rules and regulations' and generates product data. KBE is often used in the automobile and aircraft construction industry to accelerate the development and evaluation of design alternatives.
- *Case-based reasoning (CBR)*:
CBR in design supports design by reminding designers of previous experiences that can help with new situations, *Maher et al. (1995)*. CBR is a formalisation of the way designers often solve design problems: A designer recalls a relevant previous design problem and uses some aspects of the previous design as an example or a base for the new design problem. With a CBR system, the search for relevant previous designs is automated and in some occasions, also the adaptation to the new context is done automatically. Although CBR, KBE and expert systems use knowledge to solve a new problem, there is a distinct difference between CBR and the other two methods. KBE systems and expert systems usually use generalised heuristics, stored as rules of thumbs or logical inferences to solve a problem, whereas in CBR systems knowledge is stored in individual problem solving episodes (=cases). In a case, not only rules can be stored, but also documents, or drawings. CBR programs for design are among others developed for the structural design of buildings (CASECAD and CADSYN, *Maher (1995)*) and for structural design of bridges (ARCADE, *Johansson et al. (2002)*).
- *Design rationale and history systems*:
Design rationale and history systems aim at supporting the design process, within a project (less effort required to understand previous work of colleagues), as well as between different projects. Design rationale and history is usually not, or not completely documented, which often increases the effort required to maintain or re-design an artefact, or to understand previous work, *Regli et al. (2000)*. Reusing design rationale could for example:
 - o Support the design process;
 - o Act as a learning aid for designers;

- Provide justification for decisions;
- Generally, rationale and history systems are supported by other knowledge-based systems and CAD/CAM systems, or are a part of for example a KBE system.
- *Product platform:*
A product platform is 'a set of common components, modules or parts from which a stream of derivative products can be efficiently developed and launched', *Meyer et al. (1997)*. Based on a product platform a family of products can be developed, by adding one-off designed parts or modules to the common platform. The trick is to design the common parts in such a way that they do not need to be redesigned every time a new product is developed (vertical, compositional reuse). To provide for further flexibility, the platform can be made scalable, or modular. In an effective product platform design, there is a careful balance between the commonality of the individual products in the family (= degree of standardisation) and the performance (i.e. distinctiveness) of each product, *Simpson (2003)*. Examples can among others be found in the consumer electronics industry, the automobile industry and the aircraft construction industry. Shipbuilding examples that resemble a product platform approach are the German MEKO ships (Blohm&Voss), *Bohlayer (2000)*, the Danish 'Flyvefisken' class (also called 'Standflex'), *Rodholm (1990)*, or Royal Schelde's Sigma concept, *Post (2003)*.
 - *Modularisation/standardisation:*
Although modularisation and standardisation are techniques frequently used in a 'product platform' approach, modularisation and standardisation can also be applied at a more limited scale. Most standardised or modularised product parts that are not developed for a product platform are usually designed for horizontal reuse. For example: the 'Affordability through commonality' program of the US Navy, in which in a number of common naval surface ship's systems, like fire pumps, or reverse osmosis desalination modules were standardised and modularised. These modular systems were not designed for a single family of ships, but they were designed to be applied in all kind of surface ships, from frigates to Auxiliary Offshore Replenishment vessels, *Hane et al. (1995)*.

3.2 Reusing knowledge and data in ship system design

Based on the classification and the examples of knowledge and data reuse in the previous section two fundamentally different approaches to implement knowledge and data reuse in ship system design are identified:

- *design for reuse:*
This approach includes amongst others the product platform approach, standardisation and modularisation. Systems are designed not only with the present order in mind, but also with considerations to future products, to ensure reuse possibilities. Reusing the systems designed for reuse can be done with compositional reuse;
- *design by reuse:*
In this approach, system designs are generated based on previously acquired knowledge and data. The main difference with a 'design for reuse' approach is the fact that the result of the system design process can differ in every project. In this approach, both compositional and generative reuse can be applied.

Fig.1 shows a number of different scenarios to carry out the 'design for' and 'design by' reuse approaches.

To successfully implement the final scenario of an approach, the implementation of all intermediate scenarios should also be feasible. It is for example hard to imagine that it is possible to successfully automate the design process (both technically as economically), without knowledge about the effect of an engineering solution at production. However, it is not always necessary or possible to aim for the most extensive scenario. Application of an intermediate scenario is also possible. For example: follow the 'design for reuse' approach until a single technical solution and a required capacity is defined and choose per project a supplier and (accessory) component type.

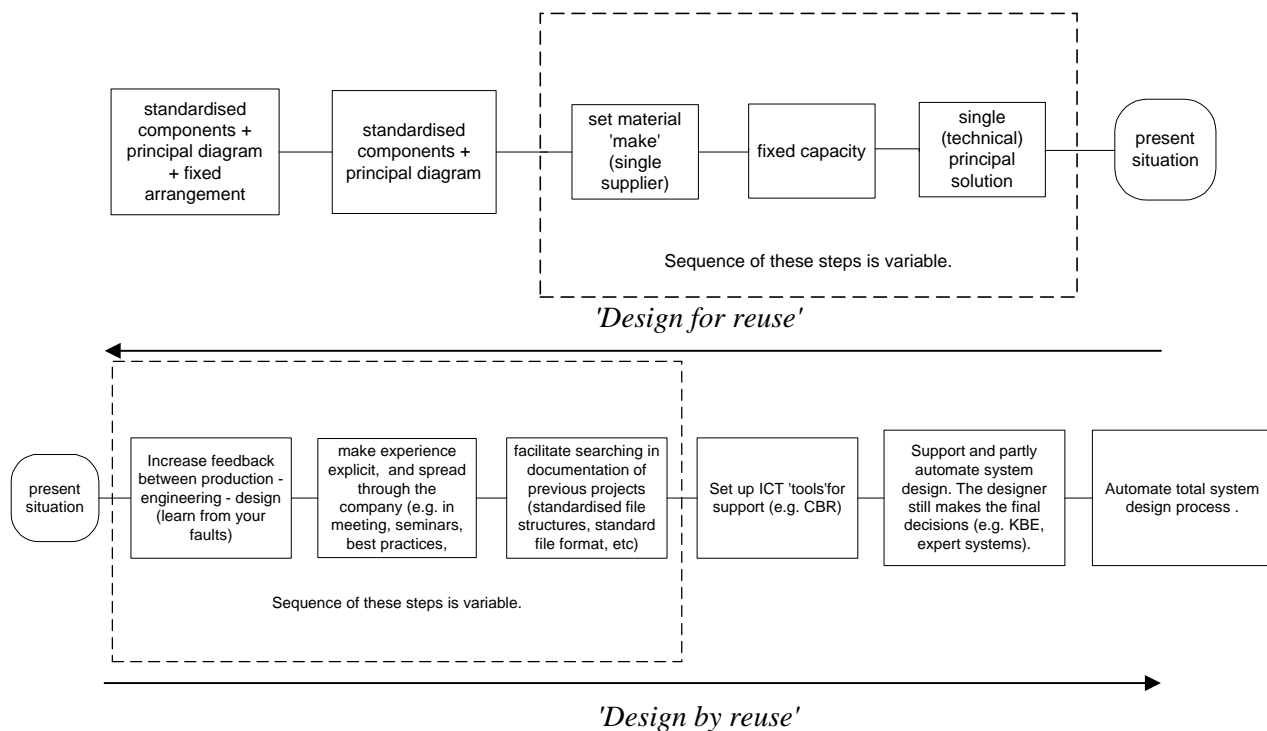


Fig.1: Different levels of 'design for reuse' and 'design by reuse' in ship system design.

In practice, combinations of the two approaches are to be expected. For example it could be possible to follow the 'design for reuse' approach up to the level of fixing the 'principal technical solution' and combine it with the design for reuse scenario 'support/automate system design', wherein than only knowledge about one type of 'principal solution' has to be incorporated. To keep a clear view, combinations of the two approaches are not further described in this paper. The effects and consequences of a combined approach can be predicted and estimated based on a composition of the results for the two separate approaches.

In order to materialize the concept of knowledge and data reuse, and to get a better idea of the consequences, problems and benefits, two scenarios have been worked out. For both approaches, the one but last level has been chosen⁸:

- *Standardise components and principal diagram ('standard system'):*
Standardise all system components ('make and type') for a family of ships. With the previous example of a BWS kept in mind, this comes down to: a single technical solution (e.g. biological treatment + vacuum collection system), sufficient capacity for the whole product family (e.g. 35-85 crew members, storage capacity for 5-10 days), a single supplier (e.g. Evac), one treatment plant type (e.g. EVAC STP75), one type of toilets and urinals (e.g. Evac 90). The arrangement of the components throughout the ship is not fixed; where to place the treatment plant in the engine room or the place and number of sanitary rooms and toilets in the accommodation is up to the designer and changes per project. In this scenario, mainly product data is reused (results of calculations, proposals, specifications, component data, etc). An example of systems designed for reuse are the systems developed in the ATC program, or the system developed for the modular engine room of Thyssen Nordseewerke, Baade (1998).
- *Support/automate the system design process:*
Support or automate the engineering process could for example be done with a Knowledge Based Engineering system (KBE) or an expert system. As long as enough knowledge can be made explicit (including domain knowledge and rationale) a system design support/automate 'tool' can:

⁸ at first sight the one but last level seems to be the maximum realizable. Results for these scenarios can be easily converted to scenarios at other levels or to scenarios based on combinations of the two approaches.

- Guide the user through the system design process, in a 'question and answer' kind of way and in the same time collect all relevant input data. For example: Crew size? What is the sailing area? Does the ship has to fulfil Marpol Annex IV requirements?;
- Automate calculations, based on the input data and calculation rules (treatment capacity, storage capacity, required number of toilets, etc);
- Automate the set-up of system configurations, based on a component library⁹, results of calculations, converted rationale, 'rules, regulations and requirements'. The program searches for suitable combinations of components with respect to desired technical solution, capacities, requirements etc, and configures a number of systems;
- Evaluate/criticize the generated solutions based on rationalised previous experiences and domain knowledge. The configuration evaluation process could be accelerated by summarising the main design results (capacities, solution type, weights, dimensions, costs) and by criticizing the solutions (for example: configuration 2 is less effective in tropical waters than configuration 1 and 3, assembly of configuration 3 requires extra attention during production due to special materials used, etc);
- Generate design documents like proposals, system specification and principal system diagram.
- Support the engineer by the conversion of principal diagrams (not related to a general arrangement plan) to diagrams at 'ship's background' (diagrams drawn in a general arrangement plan). For example by automating the calculation of pipe lengths and diameters.

In this set up, an engineer still makes the final choice which configuration to use (potentially together with the purchase department). An example of this approach for the design of pneumatic systems is given by *Korane (2002)*. This scenario involves both generative (e.g. calculating capacities) as well as compositional reuse (e.g. selecting and combining components). Till a certain extend it will also be possible to use this support/automate tool in a horizontal way, depending on how family specific the required (evaluation) knowledge is.

4. Consequences and feasibility

Which scenario is the most effective solution for ship system design, depends on the consequences, the technical feasibility and the commercial feasibility of a scenario. Although this project aims at applying knowledge and data reuse in (detailed) design, the consequences and feasibility of implementing knowledge and data reuse are evaluated with respect to all parts of the shipbuilding process, to prevent for example implementing a scenario of which the benefits for design and engineering do not outweigh the negative consequences with respect to saleability.

4.1 Consequences and feasibility of the 'standardised components' scenario

The technical feasibility of a 'standard system' approach is determined by:

- the 'application range' of a design solution:
The application range is the capacity range wherein a solution can be applied without any technical problems. If the application range is too narrow with respect to the required capacity range of a family, the number of different systems required to cover a family will be too high and the benefits of reuse will be limited. For example: can a BWS designed for a ship with a 35 crew be used for a ship with a 85 persons crew¹⁰, or is the application range more narrow and does a treatment plant only function properly in a range of ± 5 persons around its design point?;
- The component 'market-life':
The system's components should be available at the market without major changes for a reasonable period of time, to make sure that the reusable solution can also be delivered.

⁹ Component data, can be found in documentation of previous projects, supplier's catalogues, or digital catalogues like www.ZoHakuWeb.com, or www.SeaQuipment.com

¹⁰ Remaining design parameters, like sailing area or sailing profile assumed to be constant

The consequences of the scenarios can be split in two groups: measurable consequences and consequences with a 'soft' background, which are not, or very hard to quantify. This does however not mean that these consequences can be neglected. During implementation of a scenario, sufficient attention has to be paid to prevent the occurrence of these kinds of consequences.

Consequences for 'Design and Engineering':

Quantifiable:

- Decrease in engineering hours and cycle-time because of:
 - o less work: fewer calculations, no new proposals to write, no specification to be written, no offers to be evaluated, no more searching for relevant information, no more waiting at a supplier's response etc;
 - o fewer changes, less re-work, less idle time, because exact and final data are available at the beginning of the design process (this effects system design as well as detailed arrangement design).
- Fewer errors in the end result of the engineering work, especially the second and subsequent time a design is used. Correction of any errors after the first time a design is used will make it a 'proven' solution, however if any errors remain unfound not one product, but several products will be affected;
- The effort required to design and engineer a standardised solution is higher than the effort required to design and engineer a one-off;
- Lower risk.

'Soft':

- 'Freedom of design' for designers and engineers decreases, because part of the overall solution is already fixed (including interfaces with other systems/spaces): designers have to find new ways to express their creativity, less interesting work, etc.;
- There is a chance that competence and skills deteriorate, as several tasks are not executed on a regular base;
- Engineers could be unwilling to apply solutions developed in previous projects ('not invented here syndrome');

Consequences for Production:

Quantifiable:

- Production costs and cycle-time decrease:
 - o improved quality of engineering work results in a reduction of problems, mistakes and re-work during production;
 - o increased learning effects can be obtained, however learning effects can only occur when several products are build at the same location with (partly) the same workforce;
 - o improved capture, storage and reuse of production rationale, or best practices becomes relatively easy: this only has to be done for one solution. This provides possibilities for production efficiency improvements even when 'natural' learning effects cannot take place, because ships are not built at the same location, or with the same workforce.
 - o it is worthwhile to invest an additional effort in designing a 'production-friendly' design solution that saves production hours, because a design is reused a number of times (the additional effort can be spread over a number of projects).

'Soft':

- There is a chance that production employees loose certain skills as they get used to assembling only a limited number of system types.

Consequences for Purchase:

Quantifiable consequences:

- time and effort required to purchase system components decreases because it is known in advance what to buy, negotiations take place with only one or two suppliers, etc;

- for some systems, the initial component costs¹¹ of a standardised system are higher than the costs of a one-off solution, this could for example occur when components with a higher capacity than technically required are used.

Furthermore, there are a number of consequences that are quantifiable, but whereof it is hard to predict in advance whether they are beneficial or detrimental. For example: the price of components could rise when suppliers find out that they are chosen as single supplier, as they know that a purchase department has less room for negotiations. On the other hand, the opposite effect could also occur. Suppliers do not want to miss out on becoming a single supplier, so very sharp component prices are offered, recorded in a long-term contracts, which results in a price reduction compared to the present (one-off) situation. Just as with the 'social' consequences for design and engineering, the net effect will be defined by the way 'design for reuse' is implemented in the organisation.

Consequences for Suppliers:

Most consequences with respect to 'design and engineering' and 'purchase' that are identified for a yard are also valid for a supplier.

Main consequences:

- Supplier's engineering effort per system could decrease, depending on the application range of a solution and the engineering effort done by a supplier in the present situation;
- Lower risk.

Consequences for Saleability

Quantifiable consequences:

- Product costs and time-to-market decrease;
- A solution can be sold as a 'proven' solution;
- After-sales advantages for yard (spare part management in the case of integrated life services) and customers (spare part management, crew training, etc);
- A system with standardised components could be sub-optimal compared to a one-off design with respect to weight, dimensions or exploitation costs, due to the fact that the system capacity is higher than actually required;
- Standardisation can make it hard to satisfy certain specific customer requirements, such as component make.
- Significant innovations can easily be missed by sticking to a standard system design for too long, which makes the design outdated.

Once more, the effect of the consequences is strongly influenced by the way the scenario is implemented in the organisation and how the approach is sold to a customer.

4.2 Consequences and feasibility of design support/automation

The technical feasibility of the design support/automation scenario is mainly defined by the question if sufficient knowledge and data can be made explicit¹². This depends on the importance of experience in a system's design (experience is usually hard to make explicit), the number of relevant (technical) principal solutions, the number of suppliers and the relation between yard and supplier.

Consequences for Design and Engineering:

Quantifiable:

- Engineering costs and cycle-time decrease:
 - o time required to find all relevant information is strongly reduced (necessary domain knowledge, rules and regulations and rationale is captured in the program);
 - o part of the work is automated (writing proposals, evaluating offers, etc);
 - o changes can be implemented more easily;
- Higher quality of the end result:

¹¹ initial costs: costs before negotiations, or reductions due to long-term contracts etc.

¹² this includes: knowledge of engineers, knowledge of suppliers and product data

- fewer errors due to the fact that part of the work is automated;
 - more alternatives can be developed and evaluated in less time.
- To capture and store all required knowledge and data requires a substantial effort and likely also the acquisition of additional software;
- Product data is frequently subject to change, thus regular 'maintenance' (updating) of the support/automate tool is necessary;

Potential non-quantifiable consequences:

- The company becomes less reliant on the experience and knowledge of individual employees, as knowledge is captured and stored in an explicit way. Results for example in: less problems when employees get sick or resign, improved quality of work done by inexperienced employees, etc.;
- Employees might be unwilling to share their knowledge, as they have the feeling that they will lose their position.

Consequences for Production:

- Production costs and cycle time decrease:
 - fewer errors in engineering work results in fewer problems and less re-work during production.
 - Production rationale (best practices) can be made explicit and taken along in the program, to provide for 'production friendly' engineering solutions;

Consequences for Purchase:

The consequences depend on the settings of the configurator. If the program configures only one solution, 'bargain freedom' can be reduced. As long as the program configures multiple solutions, no consequences for purchase are to be expected.

Consequences for Suppliers:

Consequences:

- Suppliers have to provide product data before any requests are made, or contracts are signed;
- Suppliers should take care of keeping their product data up-to-date;
- Suppliers have to share their knowledge about system design with a yard, or part of the tool should be developed by a supplier;
- Average supplier engineering effort can decrease.

Consequences for Saleability:

Consequences:

- overall development costs and cycle-time is expected to decrease, as long as the implementation costs are outweighed by the decrease in development costs;
- changes can be implemented more quickly;
- no negative consequences with respect to saleability expected, as long as sufficient knowledge and data can be made explicit;

5 Conclusion, evaluation and further work

This paper forms a firm foundation for the subsequent work. The BWS example indicates the possibilities and consequences of reusing knowledge and data in ship system design. However, a qualitative analysis of the consequences alone is not enough to set up a decision support tool. Further, more quantitative research is required, to make a better comparison between the different scenarios possible. At a limited scale the two scenarios will be implemented in practice, which provides for an opportunity to quantify the effect of the consequences. Additional interviews with shipyard engineers will be conducted to get more quantitative data of the current situation. The evaluation of the technical and commercial feasibility in the next section indicates the areas of further research.

With respect to the technical feasibility, both scenarios have to deal with a number of challenges. The examples referred to in section 3.2 however show, that in theory both scenarios should be technically

feasible. The practical implementation of the two scenarios will outline the technical feasibility that can be achieved in practice. For example:

- minimum number of BWS required to cover a product family range¹³;
- number of suppliers that are willing to cooperate in the set up of a support/automate tool;

The commercial feasibility of an approach depends on the consequences of reusing knowledge and data for the product's selling price, the expected life cycle costs (LCC) and the time-to-market need to be quantified. A product's selling price not only depends on the total costs (cost price), but is also affected by management decisions, which could depend on factors like expected market developments, desired profit margins, strategies, etc.

The quantifiable consequences of implementing knowledge and data reuse for design + engineering, production, purchase and suppliers all have a direct or indirect effect at a product's costs and time-to-market¹⁴. The consequences of reuse for a product's saleability have to be expressed in the margin between selling price and cost price. For example: not fulfilling a special customer requirement due to application of a 'standard system' scenario, has to result in a decreased selling price or in a reduced time-to-market, to obtain equal customer satisfaction compared with a product that completely fulfils all customer requirements (*one-off*). Based on the qualitative consequences alone it is not possible to make an accurate estimation of the net effect between for example a decrease in engineering effort and a decrease in saleability. So it is hard to give a well-founded verdict about the commercial feasibility of the two scenarios. The practical implementation will help to get a quantification of for example:

- Number hours required to design a BWS based on a 'standard system' scenario;
- The influence of errors in system design at the number of changes in detailed arrangement design and production;
- The effect of errors/changes in production due to errors in system design.

Furthermore, a coupling will be made between the consequences or feasibility of reuse and system's characteristics. This relation can be used in a semi-quantitative decision support tool, to predict the consequences of reusing knowledge and data for other systems, product families or even other shipyards. Potentially in combination with the 'knowledge based engineering evaluation method' as developed by Mrs. J. Coenen and the Delft University of Technology to facilitate the estimation of required number of design and engineering hours and cycle-time for the different scenarios.

References

- AALBERS, A. (2003), *Production of complex ships*, In: Design of Marine Systems 1, Ed.: Boonstra, H., Technical University Delft
- BAADE, R.; KLINGE, F.; LYNAUGH, K.; WORONKOWICZ, F.; SIEDLER, K.M. (1998), *Modular outfitting*", J. Ship Production, Vol. 14, No. 3, pp.170-179
- BAYA, V.; LEIFER, L. (1995), *Understanding design information handling behaviour using time and information measure*, ASME Design Engineering Technical Conference, Vol.2
- BOHLAYER, W. (2000), *The MEKO principle applied to corvette design*, INEC 2000 - Marine Engineering Challenges for the 21st Century
- BONG, H.S.; HAN, S.H.; HWANG, I.W. (1994), *On the development of PROHITS: the production oriented hull information technology system for ship design and production*, 8th Int. Conf. Computer Applications in Shipbuilding, Malmö, Vol. 1, pp.47-61

¹³ To find the optimal number of systems to cover a family range, with respect to the reusability and the expected benefits the development of an optimization program could be useful.

¹⁴ The 'soft' consequences are not considered in the feasibility analysis, but should not be neglected during implementation and can affect the required implementation effort.

- BUCCIARELLI, L. (1984), *Reflective practices in engineering design*, Design Studies, Vol. 5, No. 3, pp.185-190
- CUTCOSKY, M.R., ET AL. (1993), *PACT: An experiment in integrating concurrent engineering systems*, Computer, IEEE Computer Society, Vol. 25, No. 1, pp.28-37
- DUFFY, S.M.; DUFFY, A.H.B.; MacCALLUM, K.J. (1995), *A design reuse model*, Int. Conf. Engineering Design, Prague, pp. 490-495
- HANE, T.H.; GAUTHIER, E.; LANG, S.W.; VALSI, T.J. (1995), *Modularity in H&M systems*, Naval Engineers Journal, Vol. 107, pp.149-166
- HEES, M. van (2003), *Knowledge-based computational model assembling*, SCSC 2003, Montreal
- JOHANSSON, P.; POPOVA, M. (2002), *Case-based design using weakly structured information*, Itcon, Vol. 7, pp.27-42
- KERSSENS-VAN DRONGELEN, I.C.; DE WEERD-NEDERHOF, P.C.; FISCHER, O.A.M. (1996), *Describing the issues of knowledge management in R&D: towards a communication and analysis tool*, Journal of R&D Management, Vol. 26, no 3, pp.213-230
- KLEIN, R. (2000), *Knowledge modelling in design – the MOKA framework*, Int. AI in design conference, Worcester
- KOOPS, A. (1995), *Hull form definition and computer aided design*, 5th Int. Conf. Computer Applications in the Automation of Shipyard Operation and Ship Design (ICCAS), Italy
- KORANE, K. (2002), *Pneumatics: interactive designer configures pneumatic systems*, Machine Design
- KTI, *Knowledge-based engineering and the ICAD system*, www.ktiworld.com
- LEE, D.K.; KIM, S.Y. (1996), *Object oriented approach to a knowledge based structural design system*, Expert Systems with Applications, Vol. 10, No. 2, pp.223-231
- MAHER, M.L.; BALACHANDRAN, M.B.; ZHANG, D.M. (1995), *Case-based reasoning in design*, Laurence Erlbaum Associates
- MANFAAT, D.; DUFFY, A.H.B.; LEESA, B.S. (1998), *SPIDA: Abstracting and generalising layout design cases*, Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol. 12, pp.217-22
- MEYER, M.H.; LEHNERD, A.P. (1997), *The power of product platforms: Building value and cost leadership*, The Free Press
- MOORE, C.J. (1993), *Complementary innovative computer systems for bridge design, EG-SEA-AI: Application of artificial intelligence in structural engineering*", Lausanne, pp.2-14
- NAT, C.G.J.M. van der (1999), *A knowledge-based concept exploration model for submarine design*, Delft University Press, Delft
- NONAKA, I.; TAKEUCHI, H. (1995), *The knowledge creating company*, Oxford University Press

PARK, J.H.; STORCH, R.L. (2002), *"Overview of ship-design expert systems"*, Expert Systems, Vol. 19, No. 3, pp.136-141

POST, R. (2003), *Affordability & flexibility in naval patrol vessel design*, Naval Forces, Vol.24/2

PRIETO-DIAZ (1993), *Software Reuse: Issues and Experiences*, American Programmer, Vol. 6, No. 8, pp.10-18

REGLI, W.C.; HU, X.; ATWOOD, M.; SUN, W. (2000), *A survey of design rationale systems: approaches, representation, capture and retrieval*, Engineering with Computers, pp.209-235

RODHOLM, I.B. (1990), *Standard Flex 300 – An innovation in warship construction*, Warships '90, RINA Int. Symp. Future for Surface Warships, London

SIMPSON, T.W. (2003), *Product platform design and optimization: status and promise*, ASME Computers and Information in Eng. Conf., Chicago

SMITH, J.S.; DUFFY, A.H.B. (2001), *Re-using knowledge – why, what and where*, ICED'01, Glasgow, pp.43-56

TAN, K.T.; BLIGH, T.P. (1998), *A new approach to an integrated CAD method for surface ship design*, Naval Engineers Journal, Vol. 110, No. 1, pp.35-48

THOBEN, K.D.; WOGNUM, P.M.; PELS, H.J.; BÜCHNER, A.G.; GOOSSENAERTS, J.B.M.; RANTA, M.; RANKA, A.A.M.; GIBBONS, W.M.; KERSSENS-VAN DRONGELEN, I.C. (1999), *From product data to product data and knowledge management – Requirements and research perspective*, 5th Int. Conf. Concurrent Enterprising, The Hague, pp.147-155

The Ship Domain in a Deep-Sea Area

Zbigniew Pietrzykowski, Maritime University of Szczecin, Szczecin/Poland, zbip@wsm.szczecin.pl
Janusz Uriasz, Maritime University of Szczecin, Szczecin/Poland, lubat@wsm.szczecin.pl

Abstract

The navigator steering his / her ship tries to maintain a certain area around his ship – domain – free from other objects. The shape and size of ship domain depend on a variety of factors, including the human factor, which makes the determination of the domain difficult. This is true for both restricted and open sea areas. This article deals with the outcome of the research aiming at the determination of a ship domain in an open deep-sea area. Besides, the authors describe possible applications of ship domain determined by the discussed method for situation assessment and planning a collision-avoiding manoeuvre.

1. Introduction

Hazards due to the effects of marine collisions and disasters are so serious that a lot of attention is paid to problems of navigational safety. Statistics show that human errors have caused 80% of marine accidents. One reason for this is inappropriate assessment of the navigational situation and, consequently, wrong decisions. Therefore, shipboard navigational equipment and systems are employed, among other functions, to identify dangerous situations and raise alarms. Advances in modern information technology make it possible to construct decision support systems used in sea-going ship control. One aspect of decision support consists in automatic working out of collision-avoiding manoeuvres: determination of safe ship movement trajectory, type and method of handling the ship. The criteria used for the assessment of a new situation are essential.

2. Navigational situation assessment criteria

The analysis and assessment of a navigational situation based on the assumed criteria are of critical importance for the process of decision making. The regulations applicable to a given situation are selected and prioritised on the basis of data on the present navigational situation - type of area and its specific character, encounter situation (ship, stationary objects, navigational obstructions, sea- and landmarks). The decision whether to take action or not is based on the regulations in force and adequate criteria for the navigational situation assessment. When proper actions must be taken, their kind and scope is defined.

The following criteria for navigational situation assessment can be distinguished:

- criteria resulting directly from the regulations,
- closest point of approach CPA_L ,
- safety level,
- ship domain,
- ship fuzzy domain,
- fuzzy closest point of approach.

The criterion of the closest point of approach, used in automatic radar plotting aid (ARPA), is one of commonly applied criteria of navigational safety assessment. This criterion assumes that the navigator defines a minimum (safe) distance of passing other objects (CPA_L). An additional criterion, time to the closest point of approach (TCPA), i.e. its minimum value ($TCPA_L$) is also defined by the navigator.

There are criteria that incorporate both CPA_L and $TCPA_L$ at the same time, such as the risk level which has the form of Kearon's coefficient:

$$cr = (a * CPA)^2 + (b * TCPA)^2 \quad (1)$$

where: a , b – coefficients (weights).

Another criterion, collision risk, is determined through fuzzy inference on the basis of current values of CPA and TCPA.

Pietrzykowski (2001), Pietrzykowski and Dubowik (1998) propose a method of determining the level of navigational safety based on the representation of expert navigators' knowledge using artificial neural networks with fuzzy logic. These networks, after the process of learning, enable the assessment of a situation according to the criteria used by navigators. Learning data consist of facts gathered from expert research (simulations and questionnaires): parameters characterising a navigational situation (e.g. parameters of ship state vectors) and the assessment of navigational safety level of recorded situations.

3. Ship domain

The human being intuitively tends to keep some area around him / her clear of other objects. For the navigator the analogous area is known as the ship domain. Any violation of ship domain is interpreted as a hazard to navigational safety. The shape and size of ship domain depend on a number of factors, which makes its determination difficult. The human factor is essential in this respect, both in open sea areas and in restricted waters. The concept of ship domain was introduced by *Goodwin (1975)*.

Ship domain is a area around the ship that the navigator wants to maintain clear of other vessels and objects.

The violation of the danger zone – ship domain – is interpreted as a hazard to navigational safety. The literature on the subject includes the concept of an arena of the ship. The arena covers a larger area around the ship than ship domain. The violation of the arena forces the navigator to check whether the domain will be violated if the ships do not change their courses and speeds. The earlier the violation is signalled and action taken, the more efficiently the dangerous situation can be avoided. This approach increases the number of zones around the ship. Consequently, the assessments of a navigational situation are more varied.

Two- and three-dimensional domains are proposed in the literature. The former define an area around the ship. Two-dimensional domains are shaped like a circle, rectangle, ellipsis, polygon and other complex figures. In the case of three-dimensional, spatial domains, additional parameters are taken into account, i.e. ship's draft and air draft. Their shapes are mostly those of a sphere, ellipsoid or cuboid.

Initially, the ship domain was determined by statistical methods. Ship movement trajectories are recorded as well as the areas that navigators maintain clear of other navigational objects. The domain boundary is determined on the basis of the densities of traces – trajectories – around the examined ship. Among others, domain boundaries were adopted as density lines of the trajectory ?, corresponding to the maximum value of trajectory density in a given area. Some authors consider the domain boundary as the line around a ship defining an area for which the number of recorded traces is larger than the number of traces in the case there is no domain. The application of statistical methods calls for collecting a great amount of data. Apart from the difficulties in gathering data, there is a problem of how to separate from each other particular factors affecting the domain shape and size. Moreover, the determined domains are those areas around the ship that were kept clear of other navigational objects. Those domains may not be the same as the areas the navigator wishes to keep clear of other objects (Goodwin's definition of the ship domain).

Analytical methods are based on the analytical description of domain area. The relevant methods include the determination of domain boundaries from the closest point of approach (CPA), the time to closest point of approach (TCPA), distance from the closest point of approach, as well as formulas

describing ship's manoeuvring abilities, prevailing phenomena and relations in ship movement in a pre-set area.

The literature presents analytical descriptions of the size of rectangular and elliptical domains allowing to determine the dynamic length and width of the domain. The formulas define domain dimensions as the function of geometric dimensions and ship's speed in relation to other navigational objects.

The modified version of the above approach features a relative domain for a target ship, proposed in *Smierzchalski (2001)*. The hexagonal ship domain is defined on the basis of the dynamic length and breadth determined for a given ship speed or its relative speed. Analytical methods make it possible to define a ship domain precisely. The basic difficulty, however, is that all the essential factors, including the human factor, have to be accounted for.

4. Ship Fuzzy Domain

Zhao et al. (1993) propose the concept of fuzzy boundary of ship domain. The fuzzy boundary of ship domain bounds an area determined by the ship domain boundary and a line consisting of points for which the function of membership to the set „safe distance” equals 0.5, Fig.1. In this approach only the predicted violation of the area limited by the fuzzy boundary of ship domain requires that an action be taken by the navigator.

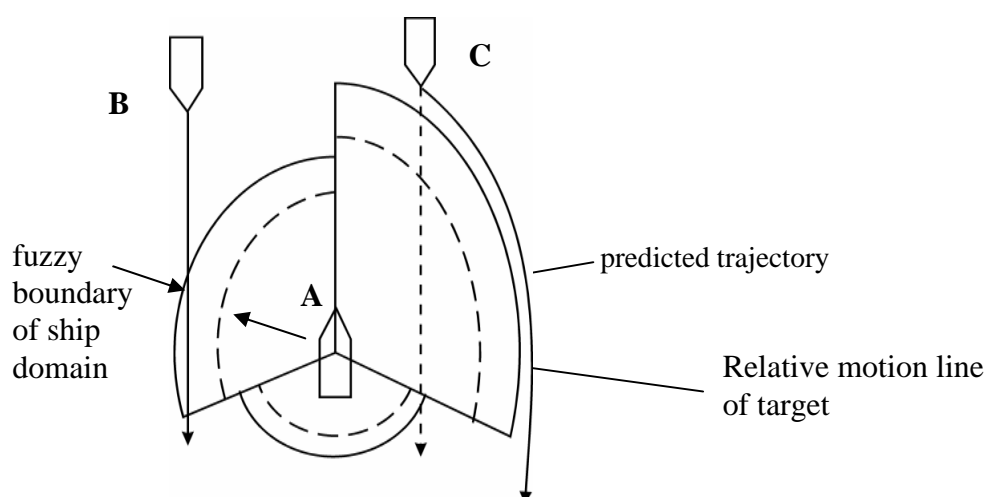


Fig. 1. The fuzzy boundary of ship domain, *Zhao et al. (1993)*

In cases of such violations the navigator has to take a proper action due to an increased risk of collision – level of navigational safety has decreased below a certain pre-set value. In the theory of fuzzy sets the navigational safety level is defined by the degree of membership of a navigational situation to the fuzzy set “safe navigational situation” or “safe navigation”.

The concepts of ship domain and fuzzy boundary of ship domain have been extended and generalised into the concept of ship fuzzy domain:

ship fuzzy domain – an area around the ship which its navigator should keep clear of other vessels and objects, the shape and size of which depend on the assumed level of navigational safety.

The ship fuzzy domain takes into account all the previously mentioned factors affecting its shape and size. The basis for its determination is the representation of the knowledge to be acquired from navigators. The knowledge is necessary for the identification and assessment of a navigational situation. This is possible through the use of artificial intelligence tools: neural networks and fuzzy systems. *Pietrzykowski (2001)*, *Pietrzykowski and Dubowik (1998)* present a method for the determination of a ship fuzzy domain in a restricted area.

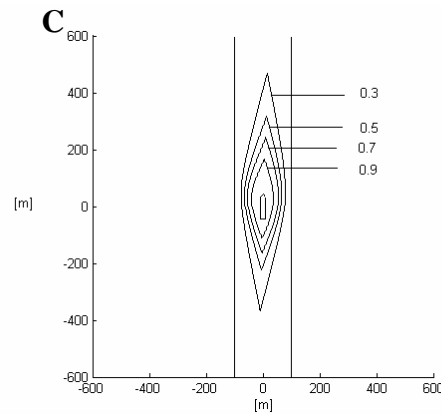


Fig.2: The fuzzy domain of the bulk carrier “Freight” (rate of turn $\omega=0$ [$^{\circ}$ /min]) for various levels of navigational safety ?

Similarly to the concept of ship fuzzy domain the criterion of fuzzy closest point of approach can be formulated. This criterion is interpreted as follows: the navigator, specifying the value of the closest point of approach CPA_L , defines a safe distance between two ships passing each other. The passing at a distance “slightly longer” will ensure a higher safety level. “A slightly lowered” value CPA_L ($CPA < CPA_L$) is acceptable as the ships passing each other at such a distance will not collide. In this way, it is assumed that there exists a tolerance interval $\langle CPA_{Lmin}, CPA_{Lmax} \rangle$, corresponding to the values of the navigational safety level ? on the $\langle 0, 1 \rangle$ interval.

The research on the methods of determining a ship domain basically focuses on restricted areas. This results from the fact that it is difficult or just impossible to apply the criterion of the closest point of approach, e.g. in narrow fairways. It seems essential to identify real criteria used by navigators for the assessment of a navigational situation in an open area. This will make it possible to determine (optimise) the ship movement trajectory in a manner reflecting navigators’ knowledge and intuition – a rational trajectory. This is important from the point of view of system reliability and, consequently, its practical use. The determined trajectory has to take into account the regulations in force.

5. The Research

The expert research dealt with the assessment of ship encounter situations in an open area in good visibility. The participating navigators – captains and watch officers had various sea service and experience. The research was carried out with the use of questionnaires. The navigators were told to determine safe distances for various encounters of own and target ships having the parameters of the m/s Freight (length overall 95.5 [m], breadth 18.2 [m], draft 5.5 [m]).

5.1. Ship Domain in an Open Area

The examined encounter situations involved ships proceeding on various courses. The obtained results were used to determine the domain boundaries for various navigational situations, Fig.3. It should be stressed that the ship domain changes in shape and size as the target ship alters its course. The dynamically changing shape and size of the fuzzy domain in a ship encounter situation may make the assessment of a navigation situation difficult.

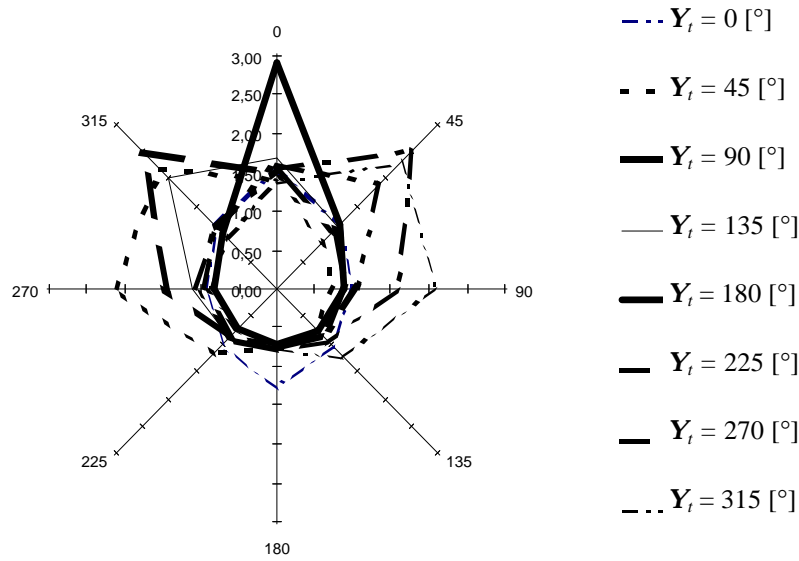


Fig.3: Ship domains in encounters for various courses of the target ship (own ship course $Y_o = 0^\circ$)

Therefore, the obtained domains were used to determine a mean ship domain D_s by statistical methods. Besides, the mean value of the closest point of approach was determined from figures given by the navigators during the research, Fig.4. The data from expert research enabled the determination of the minimum and maximum domain sizes, Fig.5.

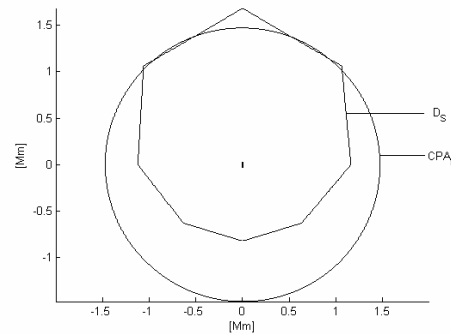


Fig. 4. Criteria for the assessment of navigational situation: the closest point of approach CPA_L and the ship domain D_s

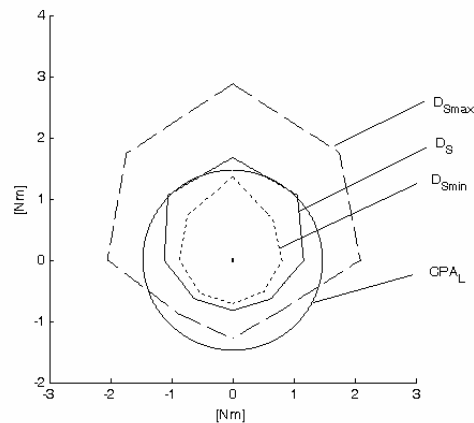


Fig. 5. The criterion of ship domain

5.2. Ship fuzzy domain in an open area

The basic problem while designing a system of navigational situation identification and safety assessment is to create an adequate knowledge base. One possibility consists in the application of artificial neural networks with fuzzy logic. These networks, after the process of learning, enable the assessment of situations according to criteria used by navigators.

The following parameters were considered as essential in the assessment of an encounter situation in an open area: distance from the target ship (d), bearing (NR) on the target ship and the target ship course (Ψ_t) relative to own ship.

In this connection, the following network structure was adopted:

- three-dimensional ($n=3$) input vector $\mathbf{x}=[d, NR, \Psi_t]^T$,
- 27 ($m=27$) inference rules,
- one-element output vector of the navigational situation assessment ($f(\mathbf{x})$).

For the presented neural network with fuzzy logic, learning data were prepared from the research results. These data consisted of, respectively, input data \mathbf{x} and corresponding to them assessments of a navigational situation, obtained by the method of psychological scaling, standardized to the $<0,1>$ interval. Then the network underwent learning process.

Figs.6 and 7 show the network responses for various values of input variables: distance from the target ship (d) and bearing (NR) on the target ship for selected target ship courses. The responses are domains for various levels of navigational safety γ . Fig.6 displays fuzzy domains where the target ship course was changed at discrete steps $\Delta\Psi_t=45^\circ$, which is equivalent to the discretization step of (non-fuzzy) ship domain. Fig.7, in turn, shows fuzzy domains where the discretization step of the target ship course was equal to $\Delta\Psi_t=15^\circ$.

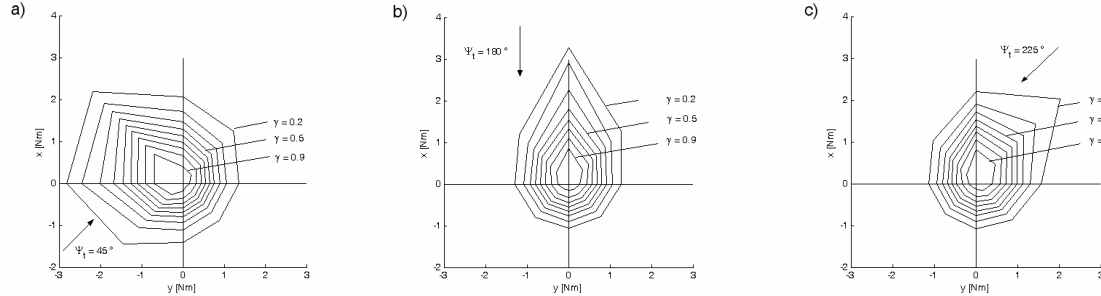


Fig.6: A ship fuzzy domain for various values of navigational safety γ for the target ship courses: a) 45° ; b) 180° ; c) 225° ; own ship course $\Psi_o = 0^\circ$; discretization step of target ship course $\Delta\Psi_t=45^\circ$

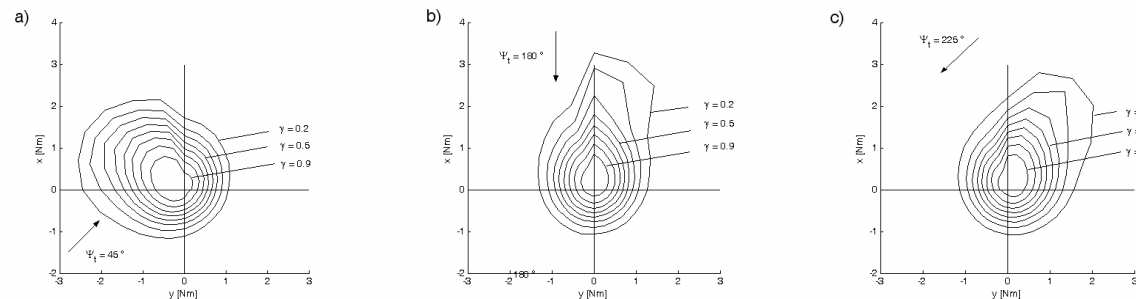


Fig. 7. Ship fuzzy domain for various values of navigational safety γ for the target ship courses Ψ_t : a) 45° ; b) 180° ; c) 225° ; own course ship $\Psi_o = 0^\circ$; discretization step of the target ship course $\Delta\Psi_t=15^\circ$

Similarly to the determined crisp domains, the shape of fuzzy domains varies depending on the target ship course relative to the own ship. In this case, too, the dynamically changing shape and size of the fuzzy domain in the encounter situation may make the assessment difficult. Therefore, like for the determined non-fuzzy domain D_S , the mean ship fuzzy domain D_{SF} was calculated. The gathered data were classified again by the method of psychological scaling. Then the network learning process was conducted. The obtained fuzzy domain for various levels of navigational safety γ is shown in Fig.8.

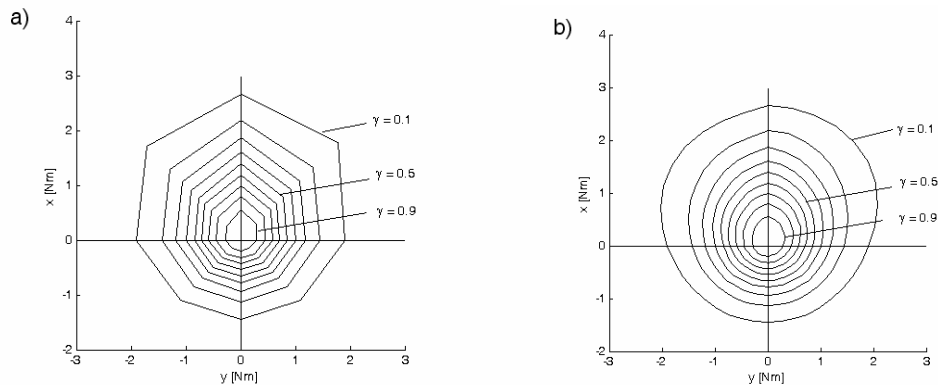


Fig.8: A ship fuzzy domain for various levels of navigational safety γ ; discretization step of the target ship course $\Delta\Psi_t$: a) $\Delta\Psi_t = 45^\circ$; b) $\Delta\Psi_t = 10^\circ$

The detailed analysis of the results leads to a conclusion that the shapes and sizes of the maximum ship (non-fuzzy) domain D_S and fuzzy ship domain for navigational safety level $\gamma=0.1$ are comparable. The minimum ship (non-fuzzy) domain is equivalent to the fuzzy domain for the navigational safety level $\gamma=0.5$.

7. Summary

Both crisp and fuzzy ship domains enable the navigator to assess the current navigational situation. Depending on the complexity of a navigational situation we can apply dynamic domains accounting for the target ship course or mean domains. Using the fuzzy domain, the navigator has additional information on the present navigational safety level. The observation of changes in navigational safety level allows to earlier identify a risk of collision. This feature makes it possible to determine a safe trajectory using various methods, e.g. optimisation, Pietrzykowski (2003a,b), Smierzchalski (2001). Besides, it enables to identify the moment of commencing a planned collision avoiding manoeuvre. The solutions proposed in this article extend the functions of collision avoidance system used nowadays on board ships and at VTS centres. These solutions allow to utilise navigators' knowledge and can be implemented in an intelligent, perhaps unmanned ship of the future.

References

- GOODWIN, E.M. (1975), *A statistical study of ship domain*, J. Navigation 28
- PIETRZYKOWSKI, Z. (2001), *Ship domain in ship's safety assessment in restricted area*, 4th Navigational Symposium, WSM Gdynia (in Polish)
- PIETRZYKOWSKI, Z.; DUBOWIK, W. (1998), *Artificial neural networks with fuzzy logic for identification of dangerous situations for ship's movement on restricted area*, Scientific Papers WSM Szczecin no 55, Szczecin (in Polish).
- PIETRZYKOWSKI, Z. (2001), *The analysis of a ship fuzzy domain in a restricted area*, IFAC Conf. Computer Applications in Marine Systems CAMS'2001, Elsevier Science Ltd.

PIETRZYKOWSKI, Z. (2003a), *Modelling decision processes in ship navigation*, 11th World Congress International Association of Institutes of Navigation IAIN.

PIETRZYKOWSKI, Z. (2003b), *Decision Models in the Vessel Movement Control Process*, 9th International Conference Methods and Models in Automation and Robotics MMAR 2003, Szczecin.

SMIERZCHALSKI, R. (2001), *On – line trajectory planning in collision situations at sea by evolutionary computation – experiments*, IFAC Conference Computer Applications in Marine Systems CAMS'2001, Elsevier Science Ltd.

ZHAO, J.; WU, Z.; WANG, F. (1993), *Comments on ship domains*, J. Navigation, No 46

Shape Modification of Hull Models in H-rep

Bastiaan Veelo, NTNU, Trondheim/Norway, Bastiaan.N.Veelo@ntnu.no

Abstract

The challenges that a designer faces in modelling shapes of arbitrary topology are well supported by the so-called H-rep concept, which is based on the tangent plane continuous interpolation of an arbitrary network of intersecting curves. Until recently, making general design changes in stages of detailed design was not well supported. A simple and efficient method is proposed to overcome this limitation in the H-rep, acting on points positioned on the surface. An effective design tool has evolved that allows feature curves to be manipulated regardless of detail, without negative effects on other design features.

1. Introduction

An effective modelling methodology for the design of non-trivial free-form shapes is the transfinite interpolation of an irregular network of curves. Recent literature refers to this methodology with the term H-rep, to indicate that it is based on the merge of two modelling concepts, namely wire-frame modelling and solid modelling. In practice, the term H-rep also implies the integration of a curve fitting/fairing algorithm, which is essential for the discussions in this article. *Koelman (1999)* gives a detailed description of the conception of the H-rep concept, and of an implementation called *fairway*TM, which is tailored to (but not limited to) the geometric design of ship hulls. Introductions to the H-rep concept and its value for the maritime industry include *Koelman et al. (2001)* and last year's presentation *Koelman (2003)*.

In essence, Koelman's implementation restores the traditional way of lines plan draughting in a computer method. To the user, the system presents itself as a curve modeller. The surface generation is kept completely under the hood, which consists of filling the mesh cells of the curve network with transfinite surface patches. These may have an arbitrary number of sides and are tangent-plane continuous across shared boundaries. Among the advantages of this modelling concept are the absence of topological restrictions on surface features, the independence of curves and their detail, and tight control over the exact shape of the surface.

Typically, when starting a design from scratch using this system, the designer starts with an initial model defined by a contour line, a deck line and a mid-ship ordinate. These curves are computer-generated, based on user-defined main dimensions. The shape of the surface patches is completely derived from the shape of the curves, so the only means to control the shape of the model is through manipulation of curves. Thus, the first step in the design process is to modify the existing curves to their correct shape, by traditional control point manipulation. When the surface is visualised at this stage, the designer will likely not be satisfied with the composite shape of surface patches. The patches are still large and the defining curves are too far apart to describe every detail that is envisioned for the design. The solution is to add more curves to the network, effectively splitting up patches into smaller ones. New curves can be generated automatically by intersecting the model with a user-defined plane, or by projecting a separate curve onto the surface. After addition of a new curve, the shape of newly split patches can be modified by manipulating the curve. With this process, the shape of a sculpted model evolves from a coarse definition to a detailed definition, until the designer is satisfied with the result.

There is a downside to this modelling methodology. As the design progresses and more curves are added to the model, more of its shape is rigidly defined. The more curves present, the smaller the surface patches, and the more local shape manipulations become. The presence of more curves also

means that the number of intersections that a curve has with other curves, increases. As a result, during curve manipulation, there is a higher risk that the curve is pulled away from these intersections, causing serious defects in the surface. Currently, no mechanism is implemented that prevents this, or that resolves the incompatibilities. Although it is possible to restore the intersections manually, by adjusting all affected curves to the changes, this is a lengthy, iterative task.

Consequently, making design changes that affect larger surface areas of the model, is discouraged at late design stages. One could say that designing sculpted shapes this way, in practice is a one-dimensional process because the design has a preference to evolve only in one direction. This paper proposes a simple and efficient method for shape manipulation of a dense curve network that is not strictly local and does not destroy the consistency and the geometric continuity of the network. This makes it possible to migrate directly from one shape variation to the other, by which the design process becomes, in our way of speaking, multi-dimensional.

2. Background

In the computer-aided design (CAD) of free-form or sculpted shapes, so-called Non-Uniform Rational B-Spline (NURBS) surfaces enjoy great popularity. NURBS surface patches derive their geometry from control points that they approximate. Whether their popularity is justified, is debatable. The challenge of designing sculpted shapes boils down to two main problems. These are the problem of geometric continuity and the problem of control.

A single NURBS surface patch without degenerate sides¹ fits only well to deformations of the square, the cylinder and the torus—of which only the torus can describe the boundary of a solid. All other geometries, thus including most subjects of design, are denoted as having arbitrary topology. For a composition of (non-degenerate) patches to describe shapes of arbitrary topology, the patches must be allowed to be organised in an arbitrary manner, where the number of patches that mutually connect with one of their corners is not always four. Maintaining geometric continuity, i.e., a smooth composite surface, is especially difficult at these irregular points.

The problem of control is implied by the fact that NURBS surface patches approximate a regular grid of control points. The problem becomes eminent, e.g., when more control points are needed locally, in order to define some local detail in a surface patch. Since extra control points can only be added in complete rows or columns, they also appear in regions where they are not wanted, because they make achieving surface fairness more difficult.

Transfinite surface patches, which derive their geometry from curves that they interpolate, do not suffer from a control problem, as the patch does not care how its bounding curves are defined. For adjacent transfinite patches to connect with tangent plane continuity, tangent information is required along the curves, which is represented by so-called tangent ribbons. In order to model arbitrary topology, the curve network must also be arbitrary, i.e., without regularity requirements². *Jensen et al. (1991)* were the first to develop a technique for the generation of tangent ribbons on such networks by using a boundary representation (B-rep³), which is a data structure used primarily in solid modelling. Their field of application was automotive styling. *Van Dijk (1994)* took this to conceptual industrial design and *Michelsen (1995)* to naval architecture.

¹ A degenerate side is a side that is collapsed to a single point. The resulting singularity makes maintaining geometric continuity difficult.

² The only requirement on curves is that they intersect and not cross each other (within some tolerance) and that they start and end at other curves.

³ A B-rep data structure describes the boundary of a solid, consisting of the three types of topological elements of “node”, “edge” and “face”. References exist in the data structure such that for each element, its neighbouring elements can be determined. The orientation of the faces, i.e., which side is facing inward or outward, is implicitly defined by the ordering of references.

Although having developed an alternative to NURBS surface modelling without the associated problems, it remained a challenge to keep the curve network simultaneously fair and consistent as a surface representation. By integrating a curve fitting and fairing algorithm, *Koelman (1999)* was able to improve that situation, and produced the described implementation for the geometric design of ship hulls, at production quality. In addition, he removed the need for the user to worry about surface patches, by following up on the suggestion by *Michelsen* to use the B-rep to its full potential as a solid representation.

In Fig.1, the hybrid nature of the H-rep is clearly visible. The nodes in the B-rep refer to intersection points in the wire-frame for their geometry. The edges refer to the curve sections in-between the intersection points and the faces refer to the n -sided patches that can be generated to fill the openings in the wire-frame. Tangent ribbons are also partly indicated.

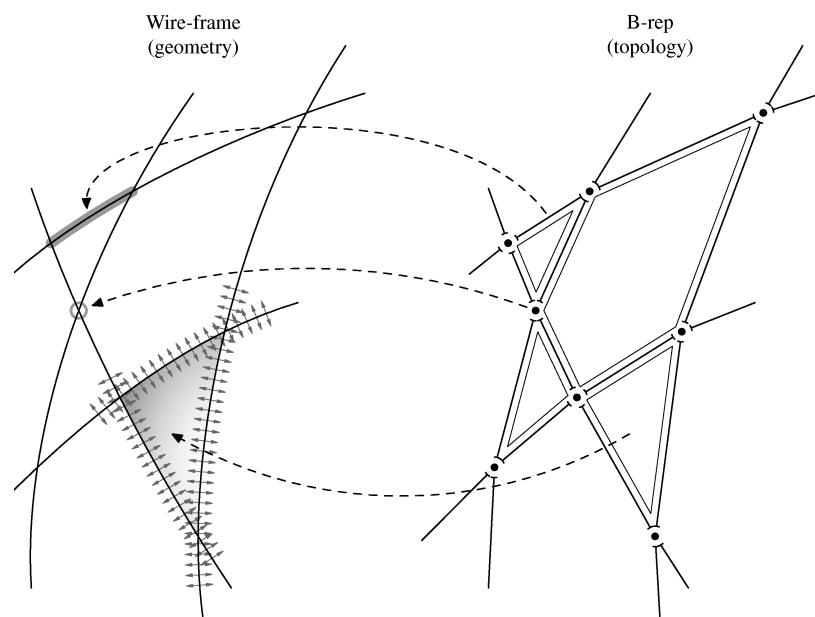


Fig.1: The hybrid representation with references between topology and geometry.

3. Related Research

The problem of the global shape of a model getting fixated by the definition of details is not specific to the H-rep and its precedents. It also exists in systems that are based on approximation of control points such as NURBS surfaces, although the consequences are not as dramatic as the surface defects that can arise in an H-rep. If a surface region is to be modified for which a larger number of control points need to be shifted, this must be done in a way that preserves the coherence between the control points, so that both the global shape remains fair and the detailed surface features are not damaged. This has been addressed by the integration of physics based properties (*Terzopoulos and Qin, 1994*), (*Léon and Trompette, 1995*) and hierarchical refinement (*Forsey and Bartels, 1988*). These references do not explicitly consider arbitrary topology however. The hierarchical refinement principle has also been proposed for the so-called surface splines (*Gonzales-Ochoa and Peters, 1999*), which approximate an arbitrary mesh of control points and thus support models with arbitrary topology. Contrary to plain NURBS surfaces, the result may be a viable alternative to the H-rep.

Free-form deformation (FFD) (*Sederberg and Parry, 1986*) is a technique to reshape a geometric model indirectly by warping the space in which it is defined. FFD is independent of the model definition, and thus competes with the method presented here.

4. Manipulation of Sets of Data Points

We will state our problem as follows. “Given a certain region on a surface that interpolates a network of curves, manipulate all curves in that region simultaneously, in a way that does not destroy the consistency of the network and does not introduce unwanted geometric discontinuities”.

We note that the details behind the process of adding a new curve consist of tracing a string of data points over the surface, as a sampling of the intersection curve or the projected curve. Then the fitting/fairing algorithm is invoked to generate a curve through these points, which is added to the model. During manipulation of curves (and thus the surface) these data points can be made to move with the curve, so that they indeed remain positioned on the surface. If we assume for the moment that all data points are persistent, meaning that they remain in existence throughout the modelling process, then the complete model can be regenerated from the data points and the B-rep alone⁴, with the help of the fitting/fairing algorithm. Thus, we can reformulate the problem as, “Given a point set belonging to a consistent H-rep, shift a selection of points to a new position, so that the distance and the direction of the shift of each individual point varies smoothly over the set”.

We will now assert our assumption. Data points that are associated with intersections between curves are persistent, because they represent the geometry of node elements in the B-rep. Currently, other data points are not persistent, as they serve no purpose after the creation of a curve. Nevertheless, in a dense curve network, there will likely be enough intersections (and thus persistent data points) to record the shape of the curves. A simple heuristic can verify this, e.g., by checking whether the number and distribution of data points belonging to a curve stands in proportion to the number and distribution of control points of the curve. If the verification fails, extra data points can be inserted at low computational cost.

4.1. Shift Vectors

Let us declare \mathbf{s}_i to be the *shift vector* for a data point i , i.e., the difference between the position of that point after and before the shape modification. We will define this shift vector as the vector sum of the sample of one or more three-dimensional vector fields.

A vector field is primarily defined by a *selection field* j of varying intensity, which is concentrated around a point, a curve or a surface, which we will call the *base* of the selection field. This base may be part of the model, or be dedicated to support the selection field. The intensity f_j of the field will be unity at its base, decreasing smoothly with increasing distance d to the base, and level off to zero at a distance r_j to the base, which we will call the *extent* of the selection field. If the base is singular, r_j is a constant; but if the selection field is based on a curve (or surface), r_j may be a function of the curve parameter (or surface parameters)⁵. In a similar fashion, we will define a vector on the base, whose length and direction may be a function of the base parameters. We will call this vector the *typical shift vector* of the selection field, denoted by \mathbf{S}_j .

In addition to a selection field, one or more *deselection fields*, enumerated by k , may take part in the definition of a vector field. Deselection fields reverse the effect of the selection field. Their definition is similar to the definition of selection fields, except that they lack a typical shift vector and their intensity g_k is opposite to the intensity of selection fields: unity outside their extent, smoothly

⁴ Actually, this is a slight simplification, as Koelman’s implementation supports curves to be composed of shorter curve sections. The end-conditions of a curve section may be dictated by the adjacent section in a *master/slave* fashion. This information can easily be conserved throughout a surface modification.

⁵ In order to guarantee that vectors in the field vary smoothly, we must require that the radius of curvature of the selection base is at least as large as its extent, everywhere. Neither may two distinct parts of the same base come closer than the sum of the extents at those parts. I.e., selection fields must not self-intersect.

decreasing in proportion to the distance d to the base inside their extent, and levelling off to zero at their base.

The vector field is then defined as the typical shift vector, evaluated on the closest point on the base, multiplied by the selection field intensity and the deselection field intensities. Especially for deselection fields it is interesting to have them act differently on the x , y and z coordinates of the vectors in the field, and thus we will redefine field intensities as diagonal matrix functions:

$$\mathbf{f}_j(d) \equiv \begin{bmatrix} f_{j,x}(d/r_j) & 0 & 0 \\ 0 & f_{j,y}(d/r_j) & 0 \\ 0 & 0 & f_{j,z}(d/r_j) \end{bmatrix} \text{ and } \mathbf{g}_k(d) \equiv \begin{bmatrix} g_{k,x}(d/r_k) & 0 & 0 \\ 0 & g_{k,y}(d/r_k) & 0 \\ 0 & 0 & g_{k,z}(d/r_k) \end{bmatrix}, \quad (1)$$

in which we have normalized the support of the intensity functions with respect to the extent of the respective field. If we then say $g_{k,y}(d/r_k) = 0$ for a deselection field based on the plane $y = 0$, we accomplish that data points in that plane will only shift in that plane and not away from it, regardless of the direction of the typical shift vector. This is advantageous if the design is symmetrical around $y = 0$ and only one half of it is being modelled.

The definition of the shift vector can now be formalised as

$$\mathbf{s}_i \equiv \sum_j \left(\prod_k (\mathbf{g}_k(d_{i,k})) \mathbf{f}_j(d_{i,j}) \mathbf{S}_j \right). \quad (2)$$

Here $d_{i,j}$ denotes the shortest distance through space from the data point i to the closest point on the base of selection field j , and \mathbf{S}_j and \mathbf{f}_j are evaluated at that position on the base. Analogously, $d_{i,k}$ denotes the shortest distance through space from the data point i to the closest point on the base of deselection field k , and \mathbf{g}_k is evaluated at that position on the base. What remains is to find suitable definitions for the selection functions f and g , and for the typical shift vector \mathbf{S} .

4.2. Selection Functions

Any function that behaves as described will give useful results. For more control of the shape of the resulting modification, one may want to vary the shape of the selection function over the base of the selection, as a function of the base parameters. This is possible in the following definition of a cubic piecewise polynomial, in which a parameter $\mathbf{k} \in [0,1]$ defines how fast the function falls off.

$$f\left(\frac{d_{i,j}}{r_j}\right) \equiv f(u) \equiv \begin{cases} \frac{\mathbf{k}^2 + \mathbf{k}u^2(u-3) + u^3}{\mathbf{k}^2} & \text{if } 0 \leq u < \mathbf{k} \\ \frac{(u-1)^3}{\mathbf{k}-1} & \text{if } \mathbf{k} \leq u < 1 \\ 0 & \text{if } 1 \leq u. \end{cases} \quad (3)$$

in which u has been substituted for $d_{i,j}/r_j$ for simplicity. This function, plotted in Fig.2, is derived from the Cox-deBoor recursive definition of B-spline basis functions. The selection function of deselection fields can simply be defined as $g \equiv 1 - f$.

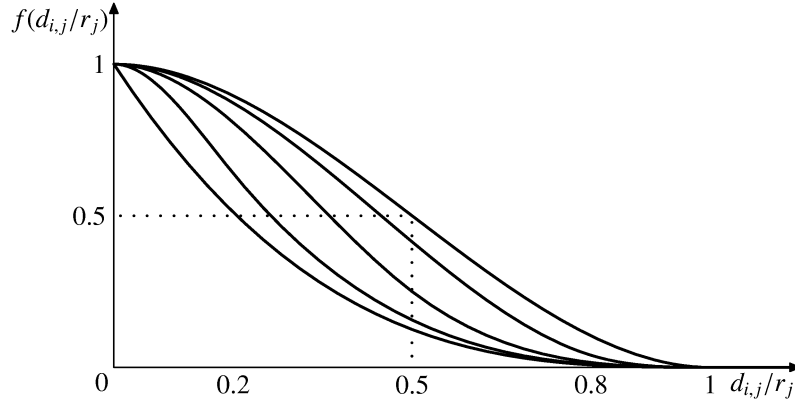


Fig.2: The selection function defined by equation (3). Smaller values of k make the function fall off faster. Plotted are $k=1.0$, which is point-symmetric about $(0.5,0.5)$, $k=0.8$, $k=0.5$, $k=0.2$ and $k=0.0$.

4.3. Typical Shift Vector

A selection field with a singular base can very well be based on a data point on the surface. It will be natural to take the surface normal at that point as the typical shift vector, scaled up or down if necessary.

For selection fields that are based on a curve, a powerful modelling tool results if the typical shift vector can be varied along the curve. Put simply, the shift vector can be defined as the difference between the base curve, say $\mathbf{c}(t)$, and an other curve, say $\hat{\mathbf{c}}(\hat{t})$. If $\mathbf{c}(t)$ is a curve on the surface prior to the shape modification, then $\hat{\mathbf{c}}(\hat{t})$ is exactly what the model will look like at this location, after the modification. Thus, designers will be able to manipulate feature curves, or even completely redesign them, while they will be able to control how the other curves (and thus the surface) in their vicinity adapts to the changes with the parameters r and k . In addition, they will be able to protect other feature curves during the modification, by basing a deselection field on them.

To make this principle work as expected, it needs to be somewhat refined. As $\hat{\mathbf{c}}(\hat{t})$ may be completely different from $\mathbf{c}(t)$, their parameterisation may not be similar. In other words, when two particles are considered, one travelling down each curve at proportional increments of \hat{t} and t , the variation in velocity of the two particles may not be parallel. The effect on the typical shift vector will be that it changes direction more often than necessary. We will remedy this by evaluating the curves with respect to arc length. In addition, if $\mathbf{c}(t)$ is a feature curve, the designer may not want to vary the complete curve, and $\hat{\mathbf{c}}(\hat{t})$ may partly coincide with $\mathbf{c}(t)$. But due to the different lengths of $\mathbf{c}(t)$ and $\hat{\mathbf{c}}(\hat{t})$, the typical shift vector may still have non-zero length in these parts, which is not intended. To counter this, we must evaluate the curves only over the curve sections that actually have different geometries.

Say that the curves $\mathbf{c}(t)$ and $\hat{\mathbf{c}}(\hat{t})$ differ from each other for $t \in [t_a, t_b]$ and $\hat{t} \in [\hat{t}_c, \hat{t}_d]$, with $t_{\text{begin}} \leq t_a < t_b \leq t_{\text{end}}$ and $\hat{t}_{\text{begin}} \leq \hat{t}_c < \hat{t}_d \leq \hat{t}_{\text{end}}$. For other parameter values, the curves coincide, although not necessarily for equal parameter values. The exact value of t_a , t_b , \hat{t}_c and \hat{t}_d can be determined by analysis of the control points and knot vectors of the curves. For a formal definition of the shift vector, we need a mapping $m: t \mapsto \hat{t}$. For a certain parameter value t_i , $m(t_i)$ must produce a \hat{t}_j so that the arc lengths of the curve sections on either side of these parameter values are proportional:

$$\frac{\int_{t_a}^{t_i} |\dot{\mathbf{c}}(t)| dt}{\int_{t_i}^{t_b} |\dot{\mathbf{c}}(t)| dt} = \frac{\int_{\hat{t}_c}^{\hat{t}_j} |\dot{\hat{\mathbf{c}}}(\hat{t})| d\hat{t}}{\int_{\hat{t}_j}^{\hat{t}_d} |\dot{\hat{\mathbf{c}}}(\hat{t})| d\hat{t}} \quad (4)$$

in which the arc length of a curve is defined as the integral of the length of the first derivative of that curve with respect to the curve parameter. Because m is inefficient to be evaluated directly, one should first assess whether arc length evaluation is at all worthwhile, by comparing internal knot spacings and control point distances of $\mathbf{c}(t)$ and $\hat{\mathbf{c}}(\hat{t})$, looking for large discrepancies. If so, the map m may be approximated by evaluating $m(t)$ at distinct values of t_i , and fitting a polynomial, say $\hat{m}(t)$, through the mapped values \hat{t}_j .

Now we are able to define the typical shift vector \mathbf{S} as the difference between relating positions on the two curves according to arc length, expressed as a function of the curve parameter t :

$$\mathbf{S}(t) \equiv \hat{\mathbf{c}}(\hat{m}(t)) - \mathbf{c}(t) . \quad (5)$$

4.4. Unintended Selections

The proposed method for shape modification is obviously simple, as we are not regarding the surface of the model at all, and only consider data points and their shortest distance to selection bases. The advantage is speed. Shift vectors can be computed quickly enough to visualise them in real time, while the designer manipulates the modification parameters. They give a sufficient indication of how the shape will be modified once the parameters are accepted. Therefore, the presented method for shape modification is interactive to a great extent.

However, in some situations this approach can be too simple. For instance, when modifying an area on the upper side of a thin wing. Because the data points on the under side are close to the upper side, they may be selected unintentionally. Even though this may be prevented by careful definition of deselection fields, there is an alternative that can be automated, which involves putting the B-rep to good use.

Table I: Pseudo-code of an algorithm that only shifts contiguous sets of data points.

```

Let  $A$  be an empty set, capable of containing node references
Mark all nodes and faces in the B-rep as unconsidered
Mark the root node as considered and add it to  $A$ 
While  $A$  is not empty {
  Compute  $\mathbf{s}_i$  for a node  $i \in A$ 
  If  $|\mathbf{s}_i| > 0$ , then {
    For all faces  $j$  that are adjacent to node  $i$  and that are still unconsidered {
      Mark  $j$  as considered
      For all nodes  $k$  that are adjacent to  $j$  and that are still unconsidered {
        Mark  $k$  as considered and add it to  $A$ 
      }
    }
    shift node  $i$ 
  }
  remove  $i$  from  $A$ 
}

```


Let us define the *root node* of a selection field as the node that references the data point that is closest to the base of the selection field. If there are several nodes that qualify, any one of these will do. The algorithm listed in Table I will only shift data points that form a contiguous selection that is rooted at the base, and prune away isolated sets of selected data points that are separated from the main selection by more than one surface patch.

5. Finishing Up

Once the fitting/fairing algorithm has re-interpolated the curves over the shifted data points, the H-rep has become a consistent and smooth modification from the original, by which we have succeeded in our objective. However, if the original primarily consisted of plane curves, such as is customary in the design of ship hulls, these may no longer be planar after the modification.

Plane curves can be restored by intersecting the modified model with the planes in which the curves were originally defined, and adding the intersection curves to the model. These new curves take over the definition of the modified shape, by which the old curves become redundant and may be removed.

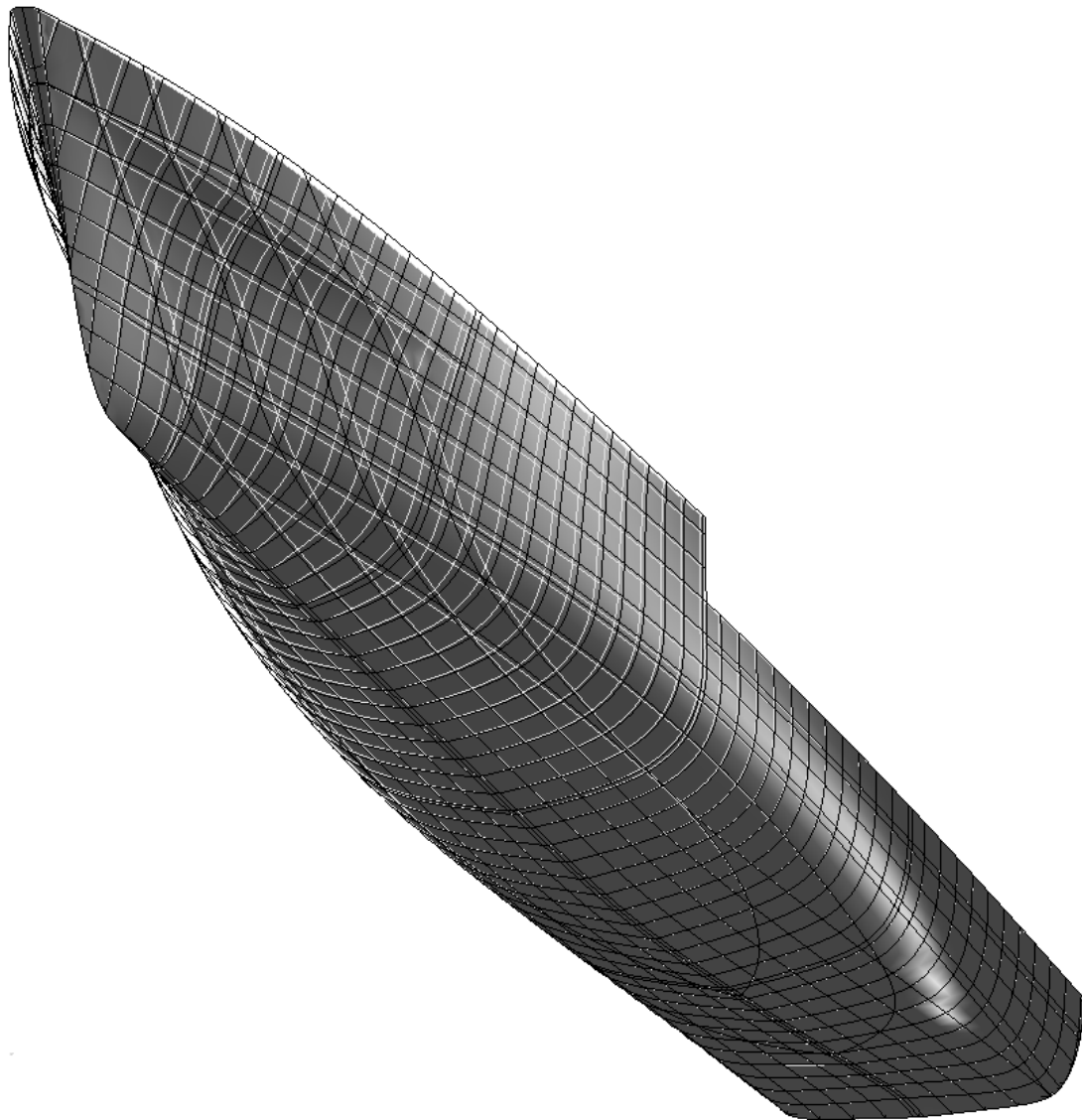


Fig.3: Modification of the hull of a frigate (shaded surface and light grey wire-frame) together with the original shape (black wire-frame overlay).

6. Applications

Fig.3 shows how the foreship of a frigate has been made slightly narrower. The selection base was a point on the hull, and the shift was in transverse direction. Note that the system has no difficulties with the knuckle line that is present in this region. A critique on this particular shape variation may be that the shell near the stem contains too much of the original shape, resulting in an extra inflection. This is due to the shape of the deselection function that was chosen to constrain the shell to the plane of symmetry, as the value of k in equation (3) was set to 1.0, see also Fig.2. A value of 0 would have been better in this regard. Optimal would have been not to base the deselection field on the plane of symmetry, but on the contour line itself. Then k could have been varied along it, 0 where the stem is sharp and non-zero elsewhere.

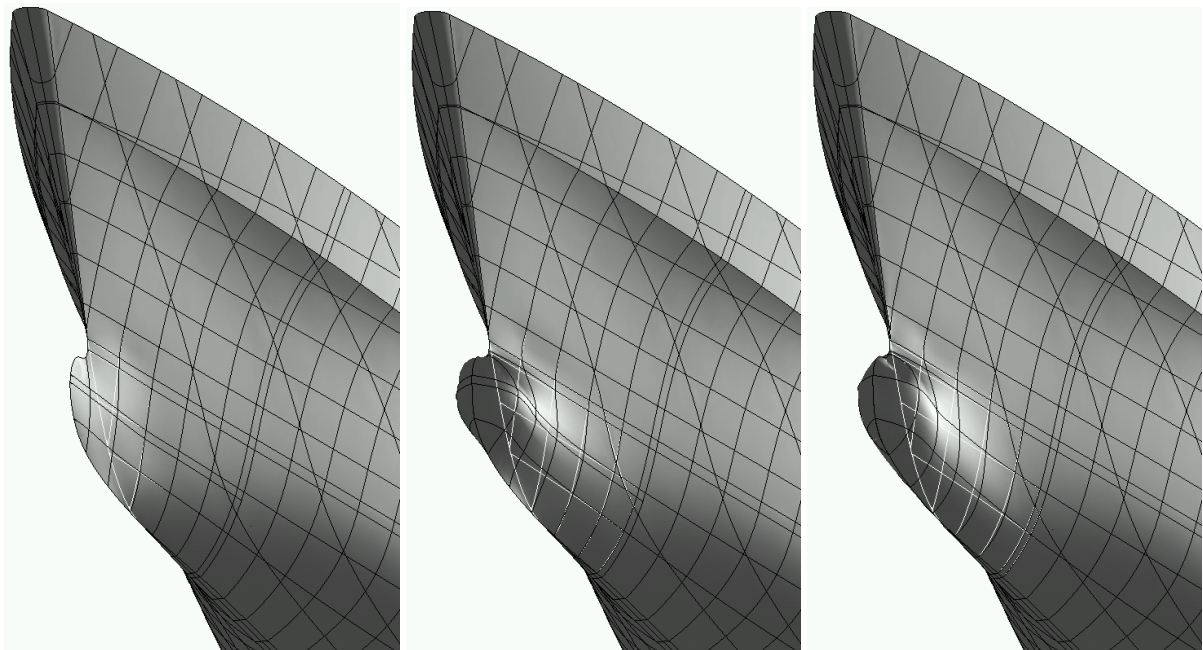


Fig.4: Three successive curve-based selections were sufficient to turn a plain bow (white wire-frame) into a bulbous bow.

Shape variations resulting from a curve-based selection are shown in Fig.4, illustrating that the method provides a powerful modelling tool. It shows all three steps in the process of designing a bulb where there was none before. Starting off with the same model as in Fig.3, the first step was to re-design the stem curve (left). An auxiliary waterline was added ending in the fore-most position of the bulb, just below the second ordinary waterline counted from below. On this line the second selection was based, giving the bulb more body (middle). Finally, the lowest waterline was dragged slightly outward, to improve the shape of the frames in the lower region (right). At the top of the bulb a dent can be seen, which is due to scarce geometric data. This was corrected with two extra frames and one extra waterline, as displayed in Fig.5.

6.1. Performance

Shape variations with selections that are based on points and planes are fast operations. However, curve-based selections can be time consuming, because computing the global closest distance of a point to a curve is an expensive operation. In the current implementation, which is not for production and serves proof of concept only, all data points in the entire model are processed. Depending on the length and complexity of the curve, the time needed for distance computations in the examples of

Fig.4 took up to several minutes on a 1.5 GHz PC. Once the distances are computed though, previewing the shift vectors happens at interactive rates.

This performance can be improved in several ways. Firstly, it may be possible to leave out several iterations of the closest point finding algorithm per data point, if the loop termination is not based on the accuracy of the closest point, but on the accuracy of the resulting shift vector. Secondly, it may be possible to disregard data points on curves that were never manipulated since they were added to the model. These curves do not actually contribute to the definition of the shape, and serve only for surface visualisation, surface quality interrogation and/or manufacturing (e.g., frame contours and the butts and seams of shell plating). In late stages of the design, there may be many of these curves. Unless the resulting shape has so much detail that the extra curves actually do contribute to the definition of the new geometry, they can safely be deleted before the shape variation and restored afterwards. A heuristic based on the typical shift vector can determine this possibility. Finally, the shape variations that are discussed here rarely involve every single data point in the model; in case it does, an ordinary affine transformation probably performs better. So computing the distance for all data points is a waste of time. Much better would be to consider data points on demand, based on the extent of the selection. For this a graph search is required, similar in nature to the algorithm discussed in section 4.4.

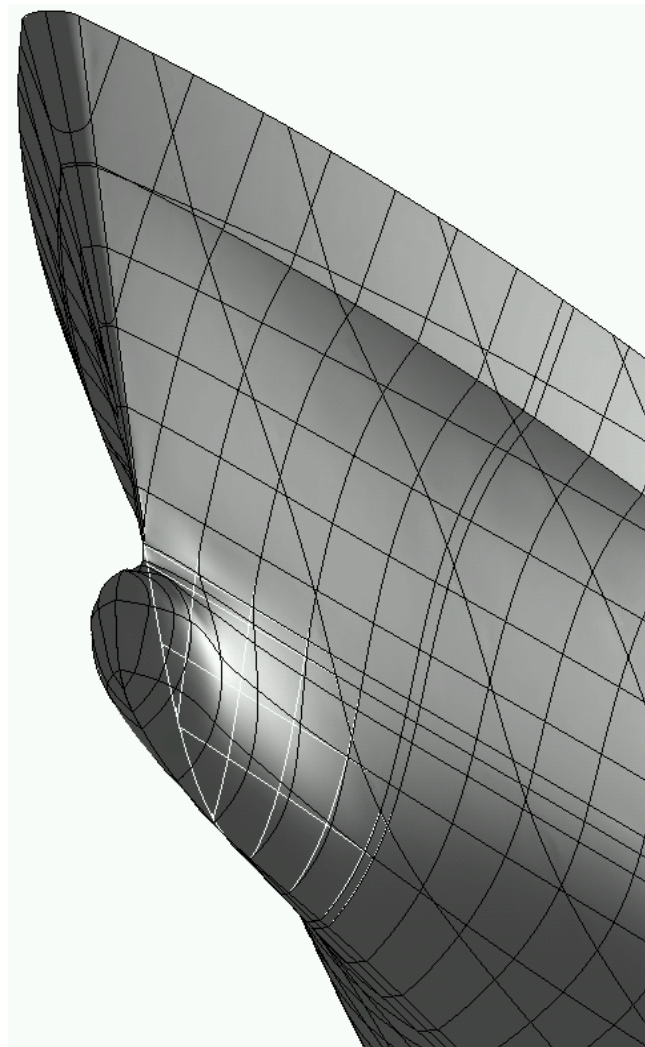


Fig.5: Extra defining data is added to ion-out the dent in the upper part of the bulb.

7. Conclusion

A simple method for the modification of a network of intersecting curves was presented, which preserves the consistency of the network, the fairness of the surface and local surface features. It has been found that feature curves may be redesigned explicitly, regardless of the detail in a design. This can be regarded as an advantage over the competing method of free-form deformation (FFD).

References

DIJK, VAN, C.G.C. (1994), *Interactive Modeling of Transfinite Surfaces with Sketched Design Curves*, PhD thesis, Delft University of Technology, Netherlands.

FORSEY, D.R.; BARTELS, R.H. (1988), *Hierarchical B-Spline Refinement*, Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques, ACM Press, pp. 205–212.

JENSEN, T.W.; PETERSEN, C.S.; WATKINS, M.A. (1991), *Practical Curves and Surfaces for a Geometric Modeler*, Computer Aided Geometric Design, vol. 8, pp.357–369.

LÉON, J.C.; TROMPETTE, P. (1995), *A New Approach Towards Free-Form Surfaces Control*, Computer Aided Geometric Design, vol. 12, no. 4, pp. 395–416.

KOELMAN, H.J. (1999), *Computer Support for Design, Engineering and Prototyping of Ship Hulls*, PhD thesis, Delft University of Technology, Netherlands, SARC B.V., Bussum, Netherlands.

KOELMAN, H.J.; HORVÁTH, I.; AALBERS, A. (2001), *Hybrid Representation of the Shape of Ship Hulls*, International Shipbuilding Progress, vol. 48, no. 3, pp. 247–269.

KOELMAN, H.J. (2003), *Exploring the H-rep Ship Hull Modelling Concept*, 2nd Int. Conf. Computer Appl. and IT Maritime Ind. (COMPIT'03), pp. 140–153. <http://www.sarc.nl/compit03.pdf>.

MICHELSSEN, J. (1995), *A Free-Form Geometric Modelling Approach with Ship Design Applications*, PhD thesis, Techn. Univ. of Denmark, Dept. of Naval Architecture and Ocean Engineering, Lyngby.

GONZALEZ-OCHOA, C.; PETERS, J. (1999), *Localized-Hierarchy Surface Splines (LeSS)*, Symposium on Interactive 3D Graphics, ACM Press, pp. 7–15.

SEDERBERG, T.W., AND PARRY, S.R. (1986), *Free-Form Deformation of Solid Geometric Models*, 13th Ann. Conf. Computer Graphics, ACM Press, pp. 151–160.

Applying Meta-Models to the Probabilistic Damage Stability

Sipke Vijver, Delft University of Technology, Delft / The Netherlands, s.vijver@wbmt.tudelft.nl

Hotze Boonstra, Delft Univ. of Technology, Delft / The Netherlands, h.boonstra@wbmt.tudelft.nl

Herbert J. Koelman, Sarc bv, Bussum / The Netherlands, sarc@sarc.nl

Erwin D. Stinstra, CQM bv, Eindhoven / The Netherlands, stinstra@cqm.nl

Abstract

The probabilistic damage stability, one of the subjects in ship design that take much computational capacity and time, determines the ship safety after a collision according to IMO regulations. One could say that it acts like a black box, of which the input are the ship form and subdivision, and the output is the attained subdivision index A. The ship designer cannot quickly estimate the value of this safety index to find out whether he is making good progress, nor can he conclude what is causing the problem whenever his ship does not comply with the requirements. An optimisation method that is based on meta-modelling should be well suited to solve the problem. A number of simulations is selected for this purpose, consisting of ship subdivisions and their corresponding attained subdivision indices. Then a meta-model is constructed which best fits the original data and which can be quickly evaluated. The next step is to study the obtained meta-model that will provide the designer with a way of estimating the index A for every possible ship subdivision and of finding the best ship in this regard. In this project we have used a ship design programme to calculate the attained subdivision index and a different programme for composing the meta-models and performing optimisations. The combination of both applications has shown that the meta-models are able to mimic the probabilistic damage stability very well. Moreover, a significant increase of the safety index can be achieved if the ship subdivision has not been optimised manually yet.

1. Introduction

The probabilistic damage stability was introduced in the 1960s and 1970s as a more flexible and more realistic alternative to its deterministic counterpart. The ship designer was supposed to be able to create a safe ship without being bothered by regulations limiting his creativity. The deterministic damage stability calculation implies that there is a maximum volume for each (watertight) compartment, whereas the probabilistic method only looks at the effects of the selected subdivision and does not prescribe any geometric boundaries.

The probabilistic calculation is used to determine the safety of a ship that has been damaged in a collision. The (presumed) effect of such an accident is that water enters the hull, which changes the distribution of weight among the ship and thus its heeling angle and stability characteristics. To examine these effects it is crucial to know the position of watertight bulkheads and decks, as they determine the boundaries of potential damages. Assuming the ship hull form has already been chosen, one can say that these positions are the main parameters in the computation of the damage stability. Their relation with the outcome of the stability calculation may be unambiguous, meaning that repeating the calculation for the same set of parameters will always yield the same result, but the designer is in fact working with a black box: he does not know exactly which processes take place inside, nor can he revert the process. This means that if he is faced with insufficient damage stability, he cannot easily find the cause of or a solution to his problem. Moreover, he will not be able to find out whether he is making good progress while working on the ship subdivision, because an early indication may be invalid at a later stage.

These observations suggest that a ship designer might welcome a little help in selecting an adequate – or even better: the best – subdivision. His only guidance now comes from his experience and instinct. Since the damage stability calculation itself does not give any clue about how close the current design

is from the theoretically best one, he will most likely accept the best he has achieved, being the first design that complies with the regulations.

The regulations we are referring to can be found in SOLAS 1974, *IMO 1992*, *IMO 1991*. Here, the probabilistic damage stability calculation is explained step by step and a requirement is formulated, being the minimum stability that a ship is allowed to possess. It can be argued that since a minimum stability value is defined, the optimum ship subdivision is the one with exactly that value. It is nevertheless very interesting to find the subdivision that provides the best stability. The difference between the theoretic maximum and the imposed minimum can then be regarded as additional freedom for the designer, who is enabled to decide which ship form and subdivision, along with cargo capacity and distribution make the best ship. Hence, we will call the ship with the highest damage stability the optimum and any compromise that the designer makes to improve other ship characteristics will be considered sub-optimal.

2. The Probabilistic Damage Stability Calculation

The principle of the probabilistic damage stability is to make an inventory of all damages that may occur in a collision and assign a probability of occurrence p to each of them. Statistics concerning accidents from the past have provided a basis for the formula that leads to this number. Next, the effects of the leakages are measured in terms of the ship survivability s ; the corresponding formula uses information from a static stability analysis.

The attained subdivision index A is used to quantify the damage stability and it equals the product of the aforementioned probabilities for all damage cases i :

$$A = S (p_i s_i) > R \quad i = 1, 2, \dots, N$$

Note that A addresses the safety; this number is based on the contributions of each damage case to the safety of the ship, rather than to the chance of sinking. A should exceed the required damage stability R , which depends on the ship length only, albeit that the number of passengers is taken into account too if a passenger ship is concerned.

The damage position and damage length are the only necessary data for the probability calculation, so it is obvious that the safety index A will need a few adjustments to make this simplification acceptable. First of all, the width (or depth) of the damage is added by means of a new factor r_i that can be described as the probability that only the outer compartment is flooded. The probability of flooding of that compartment and the next one in lateral direction equals $(1 - r_i)$ provided that only two compartments occupy half the ship breadth. A similar factor v_i is needed to predict the effect of watertight decks. In the end the equation for the attained subdivision index can be written as follows:

$$A = S (p_i r_i v_i s_i) \quad i = 1, 2, \dots, N$$

The three quantities p_i , r_i and v_i are static in the sense that they only have to be calculated once per damage case, as they depend on the ship geometry and subdivision. The stability of the ship on the other hand has a more dynamic character: in case of a collision, the damaged compartments will be flooded and this continuous process changes the ship stability from second to second. It is of course practically impossible and probably not even useful to consider the stability at every moment between the collision and the final equilibrium stage (if there is one), but somehow this flooding process has to be accounted for. The solution has been found in comparing the survivability numbers corresponding to different filling percentages of the damaged compartments. Since the weight distribution plays an important role in this calculation, it is necessary to repeat this investigation for a few different loading conditions of the ship. The weighed sum of these results equals the survivability s_i as defined above.

The assessment of the damage stability still does not look very complicated. On the contrary, many simplifications still remain. For example, leaks are currently thought to extend from the base line of the ship to their upper boundary. Other realistic scenarios in which the base of the damage does not coincide with the bottom of the ship, or in which multiple holes are observed, are not covered by the regulations, *Jensen (1995)*.

Yet, there are a few anomalies and complexities that cause the time consuming behaviour and the previously mentioned black box nature of the problem at hand, *Abicht (1990)*, *Koelman (1995)*, *Jensen (1995)*, *Björkman (1995)*:

- If one compartment is filled with water, then another may be flooded as well even if the compartments are not adjacent. The water could have entered the second compartment through a non-watertight opening or a (broken) pipeline. It is very difficult to foresee this in early design stages because many of the important details are unknown then.
- Small leaks in the ship hull may have more serious consequences than large ones. This implies that minor damages cannot be neglected in the damage stability analysis and that more calculation time will be needed.
- Intact stability requirements impose an upper limit on the vertical position of the centre of gravity of the ship. If the designer wants to avoid further restrictions on its operational conditions, he will have to comply with the damage stability regulations while keeping the centre of gravity higher than that of the undamaged ship. A few design iterations may be needed before this result is obtained.
- The number of damage cases is always very high. Usually the simplest cases (consisting of only one flooded compartment) have the largest contributions to the attained subdivision index, whereas the least likely cases (consisting of more than a dozen flooded compartments) will lead to unnoticeable increments of the index, but one cannot say in advance after how many damage cases the computation can be aborted. Note that there is no objection to ending the calculation prematurely. If a sufficient safety index has been reached after a number of damage cases has been handled, it can – in theory – only profit from the remaining cases.
- There have been reports of test results that did not come up to expectations. This way, researchers have been able to find errors and discontinuities in the probability formulas. One of the most obvious shortcomings is that the designer may come across negative values for the probability p . Since they are inherent to the probabilistic damage stability regulations, they cannot be omitted, but it is clear that a basic principle of the theory of chances is violated. This phenomenon prevents the designer from verifying whether all damage cases have been listed, because ideally $\sum p_i$ would equal 1.

3. Modelling and Optimising

In order to utilise computer power for such complex design problems, researchers have proposed optimisation methods in the quest for the best design. The first steps were taken with conventional gradient methods (for example Broyden's method) and with non-gradient methods (e.g., the Simplex method and Hookes and Jeeves). These were applied to design subjects such as the optimisation of the local hull shape and the minimisation of the construction weight, *Nowacki (2003)*, *Bertram (2003)*.

The conventional methods above have proven to be very efficient with zero or first order continuous functions, as opposed to functions of a discontinuous nature (for instance the attained subdivision index A depending on bulkhead positions). As a consequence, the so-called genetic algorithms have been introduced in ship design applications including the field of the probabilistic damage stability in

recent years, *Sommersel (1997)*, *Gammon and Alkan (2003)*, *Zaraphonitis et al. (2003)*, *Ölçer et al. (2003)*, *Guner and Gammon (2003)*, *Chen et al. (2003)*.

In 1999, one of the co-authors investigated the application of a genetic algorithm to the probabilistic damage stability, and it appeared that even though a good solution can always be found, a large number of iteration cycles are spilled with the (time consuming) calculation of impossible or irrelevant bulkhead configurations. We have therefore decided to study optimisations based on meta-modelling as an alternative. The main reasons for choosing meta-models were that this class of methods is specifically aimed at problems of which each function evaluation is expensive (meta-models should thus enable us to use our resources more efficiently) and that such models can be inspected visually or numerically, which should help the ship designer understand the behaviour of the function he is trying to optimise.

The many-to-one relation between the ship subdivision parameters and the attained subdivision index qualifies it as a function, provided that these quantities are translated to measurable parameters. The designer needs to study the behaviour of this function in order to find the best ship subdivision, but as discussed in the previous section, the probabilistic damage analysis hardly permits that. The only method available then is seemingly a comparison of alternative designs, which is how the “optimum” is currently found in practice.

There is another option to cope with this limitation, however. A replacement of the original IMO rules by a function with enough flexibility to mimic the original phenomenon is an attractive trade-off if the needed information comes within reach, *Papalambros and Wilde (2000)*. Such a replacement is known as a meta-model. To demonstrate this principle we will show how polynomials and a kriging function fit the data presented in Table I, Fig.1. More elaborated cases (with hundreds of parameters) are described in *Den Hertog and Stehouwer (2002)*, for example the optimisation of the geometry of TV screens and the optimisation of the electron gun design for picture tubes.

Table I: A random set of values of two variables x and y .

x	0.30	0.70	1.20	2.00	3.40	4.10	4.20	4.30	4.70	4.90
y	0.978	0.885	0.703	0.431	0.780	2.394	2.598	2.707	1.993	1.319

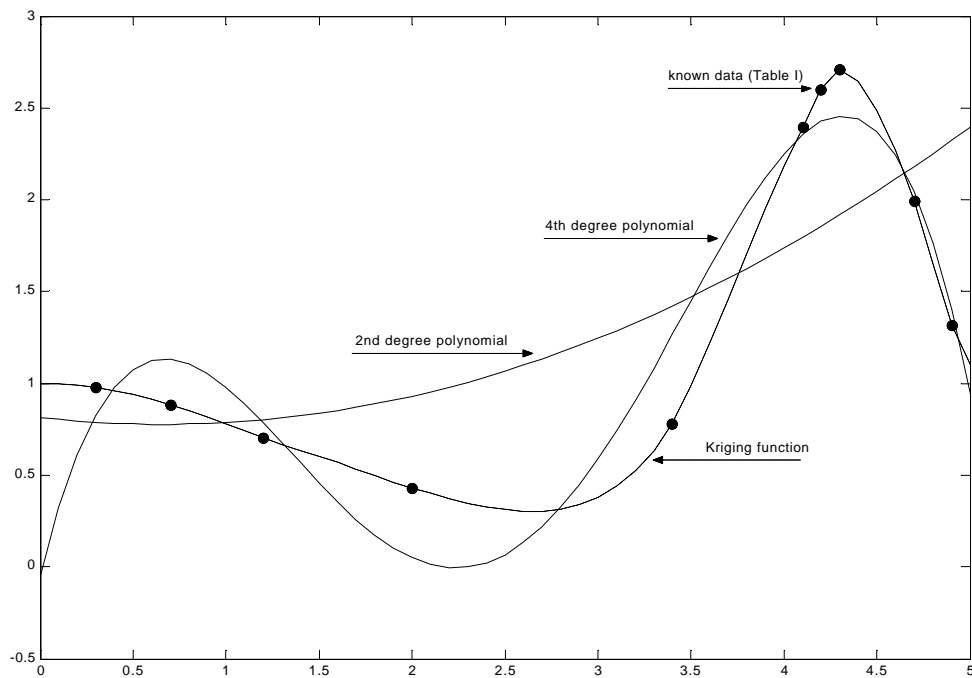


Fig. 1: Approximations of the phenomenon described by x and y by (a) a 2nd degree polynomial, (b) a 4th degree polynomial and (c) a kriging function.

Fig.1 shows clearly an important characteristic of kriging functions, named after the South African geologist D.G. Krige, namely their ability to match the given points exactly. This immediately brings us to a prerequisite for all meta-models: there should be data from which the model can be derived. Hence, we cannot avoid doing a number of expensive probabilistic damage stability calculations. We can only improve the manual optimisation of the ship subdivision if we need less computational time or fewer man-hours for it, or if we provide more information and better results while using slightly more time.

The general form of a kriging function is the following:

$$y(x) = c_0 + \sum_{i=1}^N c_i \cdot e^{\beta} \quad \text{with } \beta = -\gamma \left| x - x^{(i)} \right|^p$$

$$\text{or } y(x_1, x_2, \dots, x_K) = c_0 + \sum_{i=1}^N c_i \cdot e^{\beta} \quad \text{with } \beta = -\gamma \sum_{k=1}^K \left| x_k - x_k^{(i)} \right|^{p_k}$$

The first line corresponds to sets of points similar to that of the example of Table I, in which one variable (y) is thought to be dependent on the other (x). The coefficients c_0, \dots, c_N , γ and p have to be determined using the information of the known points $(x^{(i)}, y(x^{(i)}))$ ($i = 1, \dots, N$). The second line is applicable to situations in which the variable y depends on a number of variables $\underline{x} = (x_1, x_2, \dots, x_K)$.

We should now clearly define the problem of optimising the ship subdivision with respect to the probabilistic damage stability. It has already been explained that the only requirement is that $A > R$, and that A depends on the ship subdivision (if one assumes that KG , the number of decks and bulkheads, the trim and the block coefficient of the investigated ship are kept at a constant value). The subdivision itself is a collective term for the positions of (watertight) bulkheads and decks, which we denote by x and y (transverse and longitudinal), and z , respectively. Since we were told that sometimes the ship size is changed with the aim of improving the damage stability, we also included the main dimensions L , B and T in the list of variables. (The actual draught T_{act} was naturally recalculated every time the ship was transformed; the variable T was only used as a scaling factor.)

The next step is to select the lower and upper bounds for each variable ($x_{min} \leq x \leq x_{max}$ etc.) and to establish all interdependencies between the variables, which in our project was limited to linear relations ($\alpha_1 x_1 + \beta_1 x_2 + \omega_1 = 0$; $\alpha_2 x_1 + \beta_2 x_2 + \omega_2 \leq 0$ etc.). (Note that equalities enable us to eliminate a variable from the problem formulation.) The inequalities were particularly useful to fix requirements such as “the length of this hold should be at least 20 metres regardless of the position of the aft bulkhead”: $x_1 + x_2 + 20 \leq 0$.

The formal optimisation problem thus boils down to:

$$\begin{aligned} & \text{maximize } A = f(L, B, T, x_1, \dots, x_{N1}, y_1, \dots, y_{N2}, z_1, \dots, z_{N3}) \\ & \text{subject to } \alpha_1 x_1 + \beta_1 x_2 + \dots + \omega_1 = 0 \\ & \quad \alpha_2 x_1 + \beta_2 x_2 + \dots + \omega_2 \leq 0 \\ & \quad \text{etc.} \\ & \text{and } x_{1,min} \leq x_1 \leq x_{1,max} \\ & \quad \text{etc.} \end{aligned}$$

in which $f(L, B, T, \dots)$ is the applicable kriging function. This formulation enables us to find first and second order derivative information of the attained subdivision index function, which in turn provides the stationary points, being the local extrema for different ship configurations, *Birk (2003)*. At this stage, we only need to pick the best subdivision from the list of extrema to finish the optimisation.

4. Computer Implementation

We have used software developed by CQM to lead us through the phases described in the previous section. Compact, as this computer programme is called, distinguishes between four necessary processes on the way to the optimum ship subdivision, *Stehouwer and Den Hertog (1999)*:

1) Problem Specification

Here, the user is supposed to define the problem as has been done in the previous section, i.e. to select the objectives (response variables), the design variables and their boundaries, and to impose linear constraint functions on them.

2) Design of (Computer) Experiments (DoCE or DoE)

Next, the program decides which data points (experiments) it needs to build the meta-model upon. The selection method has four important features: (a) the number of experiments is entered by the user, so he has control over the accuracy of the model and the computational time required; (b) the evaluation scheme is space filling, in other words: the experiments are distributed over the entire feasible region as evenly as possible; (c) the scheme is non-collapsing, which means that if any of the variables shows to be irrelevant for the optimisation problem, all experiments will still provide useful information about the other variables; and (d) the design constraints are taken into account.

3) Modelling

The actual experiments (or simulations) precede the third step. A different programme is needed then, as Compact was not specifically designed for ship design problems. The simulations are typically expensive calculations and that alone could be a reason to create a meta-model. With the evaluation points at hand the unknown coefficients can be calculated. In the end we have a kriging function that is ready to use for prediction of the damage stability of unexplored ship configurations and for optimisation of the ship subdivision.

There are three methods available to estimate the accuracy of the created model:

- a) R^2 and the root mean square error (RMSE) indicate how well the model matches the original test data. Unfortunately, this is useless information in the case of interpolation models such as kriging, because they fit the test points exactly by definition.
- b) It is also possible to keep a subset of evaluation points apart and verify how well the model predicts them, but this is not a viable option because the designer wants to employ as many calculation results as possible to construct the model. It is not worth the effort to perform extra simulations solely for validating purposes.
- c) Finally, there is (leave-one-out) cross validation. All experiments are used for the benefit of the meta-model, but then the model is recreated based on all experiments but one, which serves as a testing object. This procedure is repeated until all experiments have been left out once. The CV-RMSE (cross validation root mean square error) indicates how well the degraded models can reproduce the excluded simulations; its value (a percentage) should be as close to zero as possible. Compact may need much time to do a cross validation: approximately the number of experiments times the duration of the model building process.

4) Optimisation

Only if the meta-model has been found to be good enough, the designer can proceed and finish the optimisation. The program explores the feasible region following the path dictated by the generalized reduced gradient method (implemented in the CONOPT solver code, *Den Hertog and Stehouwer (2002)*), which leads from the evaluation points to the local optima. The simulation scheme was made space filling by Compact, therefore it is reasonable to trust the programme to find all local optima, albeit that it can only find as many as the number of experiments. Usually it detects the local optima multiple times from different starting points, but it might not detect others if the followed paths do not lead to them. The more experiments

one is willing to do, the smaller the chance that optima will be missed. Finding the largest local optimum finally completes the global optimisation. It should of course be verified whether the best ship subdivision according to the programme actually results in the predicted attained subdivision index.

As said before, performing the experiments is a task for naval architectural software rather than for an optimisation tool. We have used Pias in our project, a programme consisting of modules dedicated to several types of calculations (e.g., hydrostatic and hydrodynamic analyses). The three modules required in our case were focused on hull transformations, to the configuration of the ship (by means of decks and bulkheads) and to the probabilistic damage stability, respectively.

An experiment as defined in the simulation scheme demands that the ship hull be transformed to its new size (only if the main dimensions L , B and T are considered variables). Then the subdivision has to be adapted to the new ship and especially the variable deck and bulkhead positions need be adjusted to the prescribed values. The probabilistic damage stability concludes the experiment.

Since each round is repeated a number of times, we were able to benefit from the option to externally control the program by means of macro files, a new feature of Pias. With these files the trajectory described above could be covered automatically provided the parameter values were correctly coded beforehand.

Now that we have described the software tools that will do the hard work for us, we should find a way to establish the interactions between them. The only information the designer has to provide is the ship form with its original (or preliminary) subdivision, the selection of variables and boundary values – in other words: the formulation of the optimisation problem – and the data needed for preparing the meta-model; the rest of the procedure can be left to the programmes.

As shown in Fig.2, the communication between the designer (user) and the optimisation process is in hands of a new application that at the same time takes care of the communication between the naval architectural and the modelling software. It is especially important that the simulations be performed without expecting the user to interfere, for this part is by far the most time consuming. The ideal situation for say a simulation scheme containing a hundred 15-minute evaluations is that after initiating the sequence only slightly more than 1500 minutes are dedicated to the simulations. Any input request to the user in between the simulations will result in an undesirable delay.

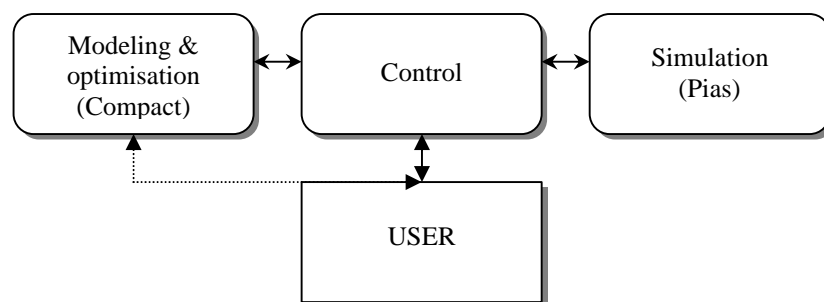


Fig.2: A scheme showing the interactions between the different programmes and the user.

The dotted arrow in Fig.2 indicates that it was not possible to manage the entire optimisation through another programme, and so the designer is now forced to switch a few times between Compact and the control application, which we have named Compias. It has been developed in such a way that the moments at which the designer is asked to take action can easily be recognised, though.

Compias first retrieves the ship size and the subdivision from the preliminary design files. Then the designer is supposed to select the decks and bulkheads whose position may be varied in order to

obtain the optimum attained subdivision index. He should also choose whether the main dimensions will be used as design variables or not.

He is then referred to Compact where he can finish the optimisation problem statement by setting the lower and upper bounds for the variables and possibly composing the constraint functions. Lastly, the simulation scheme will be made based on the number of experiments the designer is ready to do.

This list is a new input for Compias, which will now start to repeat the cycle of ship transformation, subdivision adjustment and damage stability evaluation for each item from that list. It will of course store the results of all computations and present them to the user once the task has been completed.

The attained subdivision indices are, in addition to the corresponding experiments, the evaluation points upon which the meta-model will be built. This again is a job for Compact, just as the validation of the created model and the actual optimisation of the ship subdivision. The designer can transfer the optimal parameter values to Compias and start the last simulation for verification.

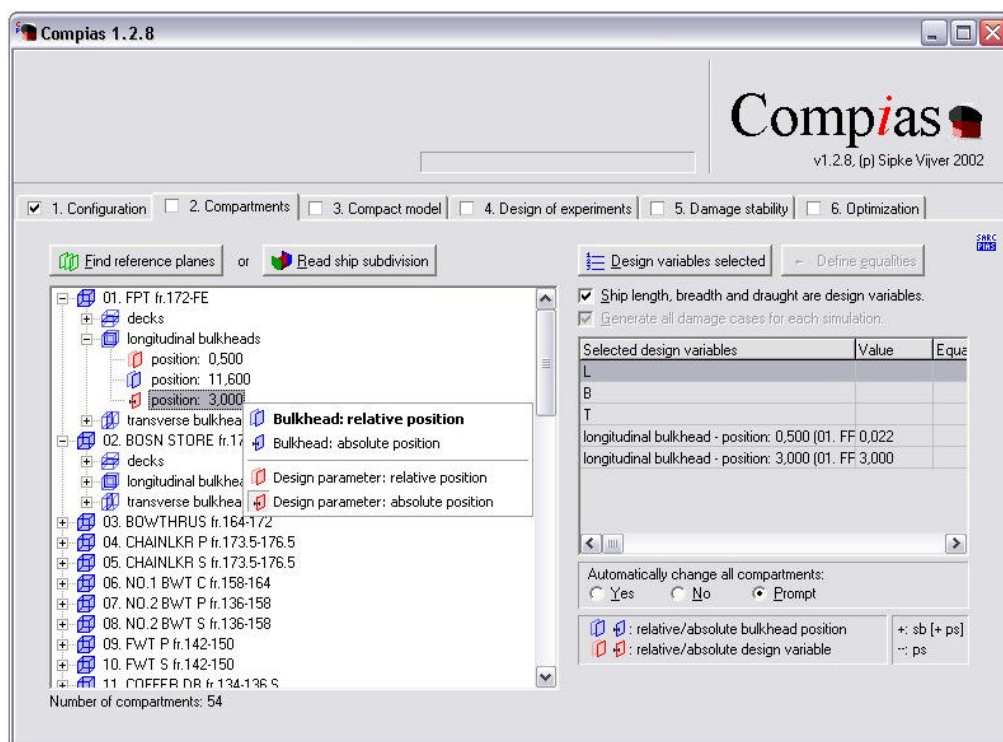


Fig.3: The graphical user interface of Compias.

5. Test Cases

We have applied our findings to an actual ship design, namely a general cargo ship of 100 metres. In the first trial we have only used two transverse bulkhead positions as variables to investigate how well the programmes worked together and to be able to present the results graphically. The selected bulkheads were named *trans01* and *trans02*; their original location was at 66.00 and 83.50 metres, respectively. We have allowed them to move approximately 6 metres back or forward, see Table II.

Table II: The selected design variables for the first test case.

Bulkhead	Original position (m)	Variation interval	
		minimum (m)	maximum (m)
<i>trans01</i>	66.00	58.25	71.20
<i>trans02</i>	83.50	77.65	88.00

Depending on the number of simulations we set Compias to do, the results varied notably. As can be seen in Fig. 4, the behaviour of the subdivision index as a function of both bulkhead positions is quite erratic and unpredictable. The larger the simulation scheme, the more complexity is revealed.

It is up to the designer to determine how detailed the kriging models should be. A rough estimation of the best subdivision can quickly be obtained, whereas a thorough study will come up with an answer after considerably more time. The latter can in general be trusted better, thus a compromise must be sought between computational costs and accuracy of the results.

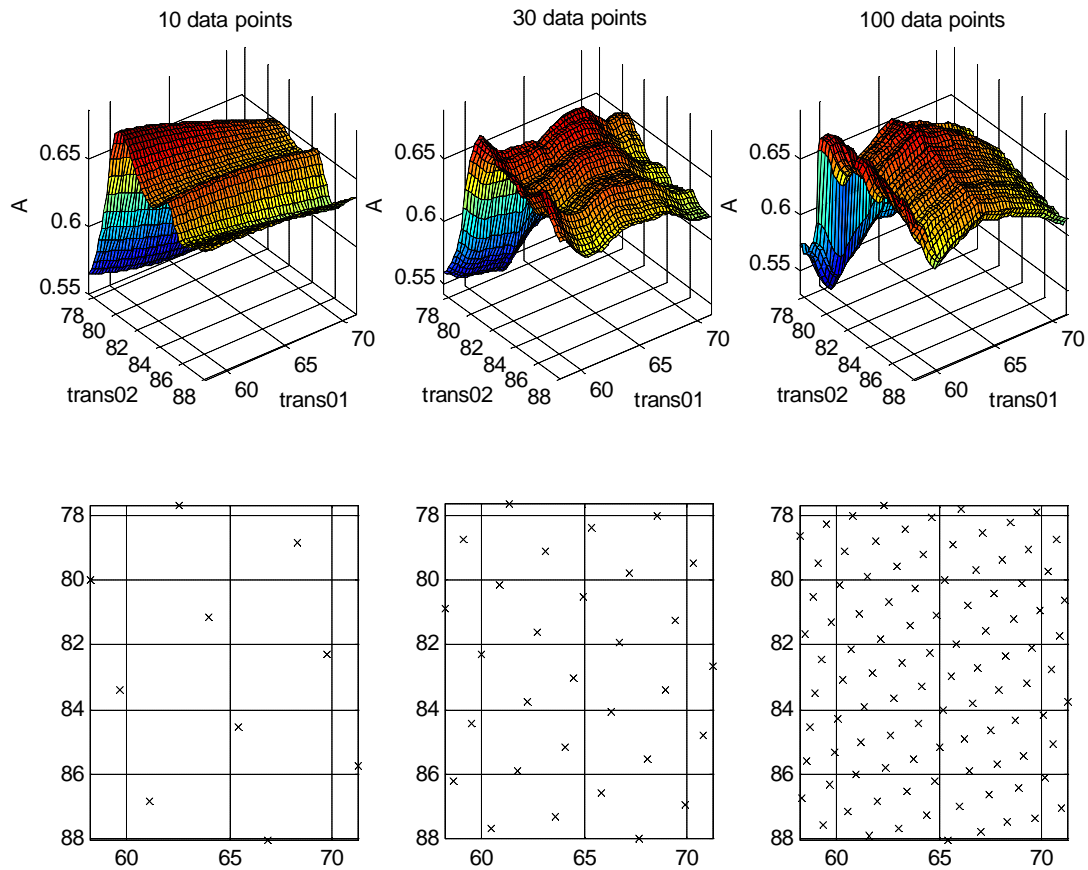


Fig.4: Different models for the attained subdivision index as a function of two bulkhead positions $trans01$ and $trans02$ (top row); the distribution of experiments (evaluation points).

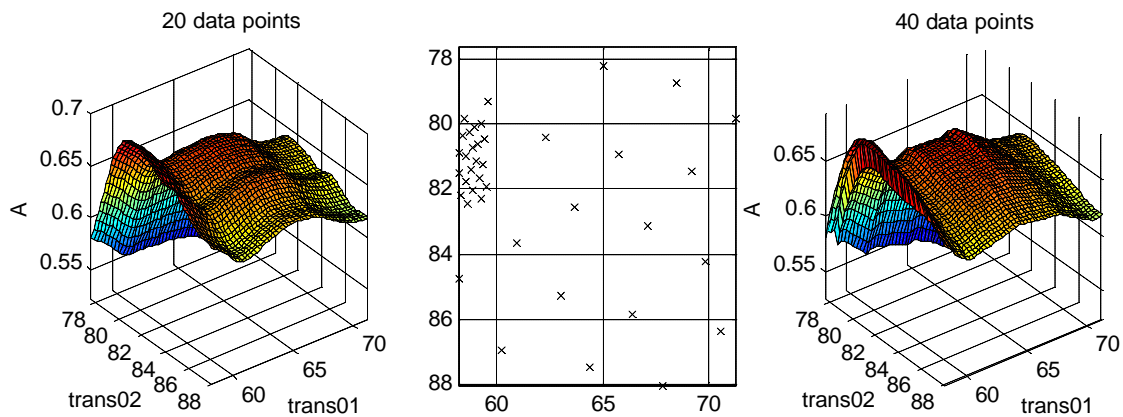


Fig.5: The index A as a function of $trans01$ and $trans02$ based on 20 simulations (left), the extended simulation scheme (middle) and the model of A based on the extended scheme (right).

One way to upgrade Compias is to do a stepwise optimisation. The programme does not explicitly support this work method, so the user will have to actively monitor the process. It implies that initially a model is composed based on a limited number of simulations and that then another simulation scheme is created to look at the interesting region(s) discovered in the first round. Fig.5 demonstrates how an extended scheme of 40 (=20+20) evaluation points improves the performance of the optimisation (see the right part of Fig.4 for comparison).

It is now interesting to see which optima the different models predict. In Table III we have collected all data from the tests with two variables:

Table III: List of predicted optima depending on the size of the simulation scheme

Simulation scheme size	Predicted optimum			Model error CV-RMSE (%)	Calculated A (-)
	<i>trans01</i> (m)	<i>trans02</i> (m)	A (-)		
10	58.25	80.35	0.687	32.06	0.686
15	58.25	83.00	0.697	18.54	0.688
20	58.25	81.15	0.701	6.78	0.689
25	58.25	81.40	0.691	19.35	0.690
30	58.25	80.90	0.689	11.80	0.689
50	58.25	80.40	0.692	7.71	0.686
100	58.25	83.55	0.697	7.71	0.686
225	58.25	87.00	0.701	3.44	0.681
20 + 20	58.50	81.15	0.693	8.54	0.689
20 + 4 x 20	58.45	82.15	0.692	3.17	0.691

It can clearly be seen that expanding the simulation scheme yields better optimum estimations, but this seems only to be valid to a certain extent: if the feasible region gets too cluttered, the models will probably have to adopt wild behaviour in order to fit all the points exactly, which will lead to poor approximations of the maximum attained subdivision index.

We have enlarged the above problem to see if Compias would be able to cope with a more realistic design challenge. We have kept the design variables from Table I and added a pair of transverse bulkheads, a few decks, two longitudinal bulkheads and the main dimensions. Two of them could be eliminated in Compias thanks to linear equality relations, for example: one of the longitudinal bulkheads was supposed to maintain the same distance to the centreline as the other. Identifying such interdependencies in advance can save much computational time. Below is a list of all variables:

Table IV: Design variables for the second test case.

Variable	Original position (m)	Variation interval	
		minimum (m)	maximum (m)
<i>L</i>	110.00	109.90	110.15
<i>B</i>	19.70	19.70	20.05
<i>T</i>	6.20	5.95	7.10
<i>trans04</i>	36.90	28.50	38.85
<i>trans05</i>	52.40	45.30	55.65
<i>trans06</i>	66.00	62.15	71.20
<i>trans07</i>	83.50	77.65	90.60
<i>deck08</i>	1.40	1.40	1.80
<i>deck09</i>	1.401	<i>eliminated</i>	
<i>deck10</i>	8.55	8.15	8.80
<i>longt11</i>	8,185	7.75	8.40
<i>longt12</i>	-8,185	<i>eliminated: longt12 = -longt11</i>	

Three simulation schemes were used for this problem: one of 100 experiments, one that contained 100 extra simulations (a stepwise optimisation was applied here), and a new scheme of 500 points.

Since none of the points of the latter list coincided with those of the other schemes, the 500 experiments could serve as a test set. In the following table we will summarize the results of the optimisation by showing the four largest local optima found along with the calculated values.

The improvements achieved by the optimisations with respect to the original (preliminary) design of the ship were a few times larger than the prediction errors shown in Table V. The results can thus hardly be called spectacular, but one should bear in mind that this test ship was a completed design. A manual optimisation on the ‘drawing board’ (i.e. one without a formal problem statement and lacking a systematic approach) had preceded our research. We suspect that an experienced designer might be capable of achieving results that are comparable to ours, albeit that we additionally learn more about other design alternatives in less time.

Table V: The four largest local optima of three different kriging models and their corresponding calculated values.

100 experiments		100 + 100 experiments		500 experiments	
predicted	calculated	predicted	calculated	predicted	calculated
A (-)	A (-)	A (-)	A (-)	A (-)	A (-)
0.644	0.646	0.664	0.667	0.689	0.660
0.644	0.652	0.664	0.667	0.685	0.662
0.636	0.660	0.664	0.667	0.685	0.662
0.633	0.632	0.664	0.666	0.685	0.662

All predictions (except one) based on the model of 200 experiments have an error of no more than 10% according to our model validations. The right part of Fig.6 shows that the cross validation relative errors (i.e. the error of a model based on all but one experiments when predicting the ignored evaluation point) are mostly between -5% and +5%. The little cloud of points on the right side corresponds to the 100 added evaluation points in the second optimisation step. The drastically decreased errors are most likely due to the higher density of simulations in that small part of the feasible region.

The left part of Fig.6 shows the difference between the predicted attained subdivision indices for the 500 test points and their calculated value. Again we can conclude that only a few exceed the error margin of 10%. Leaving out the 100 simulations of the second optimisation round gives a graph that is almost identical to the current figure, which implies that the second step mainly affects the model locally.

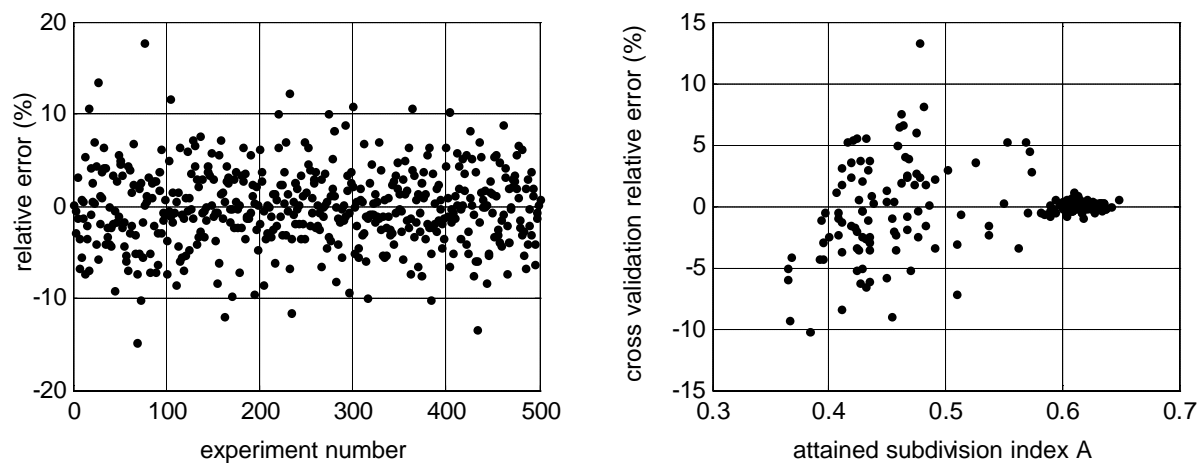


Fig.6: Prediction errors when applying the model of 200 simulations to the test set of 500 data points (left) and the cross validation relative error of all 200 simulations of that model (right).

6. Discussion

Even though the optimisations presented here did not greatly improve the damage stability of the tested ships, we were quite satisfied with the accuracy of the kriging models. An adaptation of the original ship subdivision (in order to undo the manual optimisation already performed by the designer) proved that if our formal method is applied at an early design stage, the designer will not only save much time but also have a useful tool to support his decisions concerning the position of bulkheads and decks.

We claim that the probabilistic damage stability will be less time consuming if Compias is used, because the designer only needs to set up the calculations and then leave the work to a computer that can run 24 hours a day. Of course one should consider that several aspects of the simulations have a great influence on the duration and the efficiency of the process. It means that the optimisation can be controlled to a certain extent. Some of these aspects are for example the number of variables, the size of the feasible region, the maximum number of simultaneously damaged compartments considered and the number of simulations allowed. In our cases an experiment could take 15 minutes up to two hours (on modern computers), which is a significant difference, especially if hundreds of experiments are planned. Regarding this issue we have concluded that an enhanced version of Compias should support distributing the simulations over a network of computers.

Another improvement of the programme, albeit from a different perspective, would be to really integrate all components that require user input, so that he does not need to switch between programmes. We even picture a situation in which Compias is united with the naval architectural programme, because the most important function of the former is to control the latter, which can probably best be realised from within.

Note that the number of variable bulkhead and deck positions is fixed as soon as the optimisation problem has been formulated. It is therefore not possible to start a design with an empty ship and to let Compias (in cooperation with the other programmes) determine the overall best subdivision, unless the designer is willing to repeat the optimisation procedure for different numbers of variables. This would still require a more or less elaborated design, i.e. a ship in which the bulkheads are placed within the acceptable limits. An important part of the designer's job is to impose those boundaries on the system and to avoid conflicts in the compartment definitions caused by for example overlapping spaces. The naval architectural software will notice these errors, but it cannot correct them on its own and it will thus eliminate the simulation process by demanding user input.

Finally, we would like to emphasise that the results of stepwise optimisations have been very good so far. It is advisable to follow this path when designing the ship subdivision, and a future version of Compias should incorporate this option.

References

ABICHT, W. (1990), *The probability of compartment and wing compartment flooding in the case of side damage - New formulas for practical application*, STAB'90: 4th Int. Conference on Stability of Ships and Ocean Vehicles, Naples, pp.117-124

BERTRAM, V. (2003), *Optimization in ship design*, Optimistic - Optimization in Marine Design (BIRK, L.; HARRIES, S. (eds.)), Mensch & Buch Verlag, pp.27-52

BIRK, L. (2003), *Introduction to nonlinear programming*, Optimistic - Optimization in Marine Design (BIRK, L.; HARRIES, S. (eds.)), Mensch & Buch Verlag, pp.53-82

BJÖRKMAN, A. (1995), *On probabilistic damage stability*, The Naval Architect, pp.484-485

CHEN, S.; WANG, L.; WU, X.; WANG, X. (2003), *Multi-objective genetic algorithms and their application to ship fleet optimization*, IMDC '03: 8th Int. Marine Design Conf., Athens, Vol.II, pp.319-327

DEN HERTOOG, D.; STEHOUWER, P. (2002), *Optimizing color picture tubes by high-cost nonlinear programming*, European J. Operational Research 140, pp.197-211

GAMMON, M.A.; ALKAN, A. (2003), *Initial vessel design by evolutionary optimization*, IMDC '03: 8th Int. Marine Design Conf., Athens, Vol.I, pp.77-88

GUNER, M.; GAMMON, M.A. (2003), *Optimization of total propulsive efficiency in a propeller-stator combination using an evolutionary algorithm*, IMDC '03: 8th Int. Marine Design Conf., Athens, Vol.II, pp.55-66

IMO (1991), *Resolution A684(17), Explanatory notes to the solas regulations on subdivision and damage stability of cargo ships of 100 metres in length and over*, London

IMO (1992), *Chapter II-1, part B-1: Subdivision and Damage Stability of Cargo Ships*, SOLAS Consolidated Edition 1992, Consolidated Text of the International Convention for the Safety of Life at Sea, 1974, and its Protocol of 1978: Articles, Annex and Certificates, London, pp.89-100

JENSEN, J.J. (1995), *Damage stability rules in relation to ship design*, WEMT '95 (West European Conference on Marine Technology): Ship Safety and Protection of the Environment from a Technical Point-of-view, Copenhagen, pp.71-96

KOELMAN, H.J. (1995), *Damage stability rules in relation to ship design - freedom is just another word for nothing left to lose*, WEMT '95 (West European Conf. on Marine Technology): Ship Safety and Protection of the Environment from a Technical Point-of-view, Copenhagen, pp.45-56

NOWACKI, H. (2003), *Design synthesis and optimization – A historical perspective*, WEGEMT'03: Optimistic - Optimization in Marine Design (BIRK, L.; HARRIES, S. (eds.)), Mensch & Buch Verlag, pp.1-26

ÖLCER, A.I.; TUZCU, C.; TURAN, O. (2003), *Internal hull subdivision optimisation of ro-ro vessels in multiple criteria decision making environment*, IMDC '03: 8th Int. Marine Design Conf., Athens, Vol.I, pp.339-351

PAPALAMBROS, P.Y.; WILDE, D.J. (2000), *Principles of optimal design - Modeling and computation*, 2nd ed., Cambridge University Press

SOMMERSEL, T. (1997), *Application of genetic algorithms in practical ship design*, IMDC '97: 6th Int. Marine Conference, Newcastle, pp.611-626

STEHOUWER, P.; DEN HERTOOG, D. (1999), *Simulation-based design optimisation: methodology and applications*, 1st ASMO UK / ISSMO Conf. Engineering Design Optimization, Ilkley

VIJVER, S (2003), *Doordringen tot de Probabilistische Lekkerekening*, MSc thesis, Delft University of Technology

ZARAPHONITIS, G.; BOULOUGOURIS, E.; PAPANIKOLAOU, A. (2003), *An integrated optimisation procedure for the design of ro-ro passenger ships of enhanced safety and efficiency*, IMDC'03: 8th Int. Marine Design Conference, Athens, Vol.I, pp.313-324

Ship Propulsion Control: A Fresh View

Hugo Grimmelius, TU Delft, Delft/The Netherlands, h.t.grimmelius@wbmt.tudelft.nl

Douwe Stapersma, RNLNC, Den Helder/The Netherlands, d.stapersma@kim.nl

Paul Schulten, RNLNC, Den Helder/The Netherlands, p.j.m.schulten@kim.nl

Abstract

A research project with Wärtsilä/Lips and MARIN intends to gather and apply knowledge towards more advanced propulsion control. Apart from a fresh look at control strategies, also determining the control objectives is discussed, considering alternatives to the shaft speed as controlled variable.

1. Introduction

There are several reasons for not being fully content with the traditional shaft speed control of propulsion. First of all there is the load carrying capacity of the driving engine. Gas turbines and electric motors can develop almost full power over a wide speed range. But modern highly turbocharged diesel engines (from 4-stroke high-speed diesels in naval ships to 2-stroke low-speed diesels in large containerships) cannot even develop full torque at reduced speed. This prevents optimal acceleration performance, and causes thermal cycling loads on the engine in heavy seas. *Grimmelius and Stapersma (2000,2001)* were inspired by non-satisfactory behaviour of propulsion systems on board of frigates and proposed a method to quantify thermal load for a diesel engines. These papers showed that engine speed disturbances as caused by seaway could be better suppressed by pitch control than controlling the fuel rack. A more liberal use of pitch control could improve things a lot, but present Class rules forbid to do so and current pitch actuating mechanisms are not designed for that job. *Van Spronsen en Toussain (2001)* proposed a radical solution for the problem. He investigated the full integration of the engine speed controller with the pitch controller and optimised the resulting Multiple Input - Multiple Output (MIMO) controller with modern H ∞ techniques. *Grimmelius and Stapersma (2003)* compared four control strategies, i.e. fuel rack control of shaft speed, pitch control of shaft speed, MIMO control of shaft speed, and no control of shaft speed for command tracking, disturbance rejection and limiter behaviour. Criteria were the thermal load of the diesel engine and wear of diesel engine and of pitch actuating mechanism caused by mechanical loading. In order to do so all three load and wear aspects were quantified by introducing predictors based on first-principle physical insight. *Schulten et al. (2004)* investigated a classically governor-controlled two-shaft diesel driven installation during a turn. This research included a full hydrodynamic manoeuvring model of the ship and a method to predict the difference in overloading of the propellers as caused by transverse velocities during the turn. Until now only engine speed has been used as controlled variable (input to the controller), because this signal is readily available and its associated important limitations (maximum and minimum engine speed). But technology permits to use alternatives: ship speed, torque, or even thrust (if it can be measured) or a derived variable based on several sensors could be used in future, e.g. a characteristic propeller quantity. The latter is inspired by another, typically naval requirement: cavitation avoidance. It would be a challenge to minimize cavitation not only in certain well defined nominal conditions, but in real operational practice (including manoeuvring and seakeeping). Although keeping the pitch at its nominal value has been the simple strategy up to now, research has revealed that perhaps a clever pitch control in accordance with the actual shaft and ship speed could improve performance. These thoughts and the experience of *Grimmelius and Stapersma (2003)* led to a belief that a more structured approach was required to develop a fresh view on propulsion control.

2. Analysis

2.1 Propulsion system

A ship propulsion system in general can be depicted in a block diagram, Fig.1. At the r.h.s. the force

balance between ship resistance and propeller thrust provides a net force that, after division by ship mass (including entrained water) results in an acceleration that can be integrated to obtain ship speed. Ship resistance depends on ship speed and so the integrator feeds back into the resistance block. The result is a first-order type of loop although the system is non-linear, because ship resistance depends non-linearly on speed. Also the water entrained by the hull adds to non-linear complexity, this less important effect is neglected here. On the l.h.s. of Fig.1 the torque balance between engine output (brake) torque and propeller torque results in a net torque which after division by the effective shaft inertia (i.e. summing up the rotational inertia of engine, transmission and propeller, corrected for the several rotational speeds n that result from the gear ratios in the transmission line) produces angular acceleration which is integrated to obtain shaft speed. Shaft speed feeds back into the engine block since engine torque generally is dependent on shaft speed.

The ship speed after correction for the wake factor provides the entrance velocity of the propeller. The entrance velocity coming from the right together with the shaft speed coming from the left can be divided to yield the flow angle β or its tangent, the advance ratio J . Division of the two state variables shaft speed and ship speed is a major non-linearity. Both propeller thrust coefficient K_T and torque coefficient K_Q are a function of flow angle or advance coefficient J . Both coefficients are a function of propeller pitch which may vary in case of pitch control. For obtaining propeller thrust and torque itself, the thrust and torque coefficient must be multiplied by shaft speed squared. This multiplication is another non-linearity of the system.

The propeller pitch is set by a pitch control and actuating system which normally is a hydraulic servo system getting its setpoint from the ship's propulsion control system. The propulsion control system also provides a setpoint for the shaft speed to the diesel engine block which contains a block for the engine governor.

The main disturbances that will act on the system is on the r.h.s., i.e. at the ship side. The ship resistance increases in a heavy sea. Although the static effect on the engine load line can be severe, Fig.1 shows that this disturbance finds the ships mass and its associated integrator blocking the way. In fact this integrator acts as a "low pass" filter between the disturbance amplitude and its effect on the diesel engine dynamic load. The second disturbance however acts at a point after the ship's translation integrator: seaway and the resulting ship's movements will cause amplitudes for the entrance velocity of the propeller. In Fig.1 this disturbance acts on the wake factor. From here a dynamic disturbance can find its way to the engine more directly: the entrance velocity changes the attack angle of the flow relative to the propeller blade. This changes propeller torque and disturbs the shaft rotational balance which through the integrator causes a dynamic change of shaft speed. This is sensed by the engine and in particular by the governor which immediately tries to restore the balance by adjusting the fuel injection and thus delivered torque. Thus only the shaft dynamics can block the wake disturbance from actually finding its way to the heart of the diesel engine.

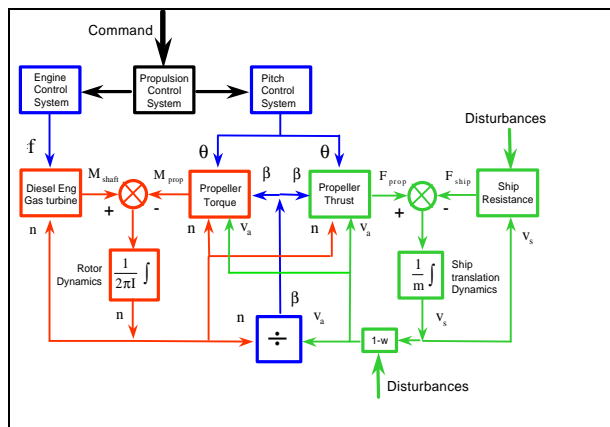


Fig.1: General block diagram ship propulsion system

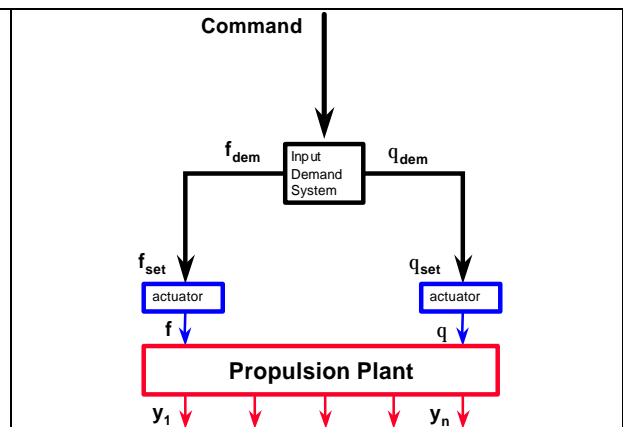


Fig.2: I/O block diagram of propulsion plant with fuel rack controlled diesel engine and CPP

2.2 Propulsion plant with only feed forward control

The basic propulsion plant of a ship driven by any propulsion engine or motor and a controllable pitch propeller (CPP) has two *controlling* variables as inputs, Fig.2. For a diesel engine as the propulsion engine the fuel rack f is the controlling variable (for a gas turbine this would be the throttle position and for an electric motor the actuating field current). The other controlling variable is the pitch (q) of the CPP. In state space description, *Skogestad and Postlethwait (1996)*, the controlling variables form the input vector $[u]$ with m elements, which in our case therefore consists of two elements or $m = 2$, i.e. vector $[f, q]$.

Normally there is an actuator or a servo system between the set point and the actual value of a controlling variable. These actuators are considered separately and act as the output part of the control system. For simple operation single lever control is required. Then a converter between the single lever command and the demand signals for the controllers must be provided, the input demand system. In Fig.2 (no feed back controllers, only actuators), the demand signals are identical with the setpoints to the actuators. In *Grimmelius and Stapersma (2003)* this system was referred to as "no control", meaning "no feedback control". The input demand system together with the actuating systems however form a sort of rudimentary feed forward control system which at least offers the luxury of a (pre-programmed and therefore non-intelligent) single lever control and the possibility to schedule the pitch (combinator curve).

The propulsion plant has many outputs that offer itself as *controlled* variable: shaft speed, torque, thrust, ship speed etc. In state space description the controlled variables form the output vector $[y]$ with n elements, i.e. vector $[y_1 \dots y_n]$. Any output variable that will be used for controlling purposes must of course be measurable. But ideally there must be other good reasons as well for selecting an output as controlled variable. Shaft speed can easily be measured, but apart from a requirement to keep its value within limits, there is nobody waiting for shaft speed control as already argued in *Faber (1993)*. Moreover with two controlling input variables it is perhaps possible to have control over more controlled output variables than just one.

2.3 Single output control

When a single variable (y_1) is selected as controlled variable its actual measured variable is compared to the demand value of the input demand system and the error is then given to a controller, Fig.3. The controller can be located either in front of the fuel rack set point or the pitch set point.

For fuel rack control (l.h.s. of Fig.3) and if the controlled variable is the engine speed ($y_1 = n$) we have the normal engine speed governor. With the input demand system the combination of engine speed and propeller pitch can be predefined (combinator curve). The terminology "fuel rack control" or "shaft speed control" for a governor both are only partially true and therefore misleading, since the word "control" is used in two different ways. In "shaft speed control" it refers to the shaft speed being the *controlled* variable while "fuel rack control" indicates that fuel rack is the *controlling* variable. "Fuel rack control of shaft speed" avoids this ambiguity but is still dangerous language. In fact it is wise to reserve the word "control" for the *controlled* variable and not use it in conjunction with a controlling variable. "Shaft speed control by fuel rack" then is the correct wording for an ordinary speed governor on a diesel engine.

The case where the controlled variable is still the engine speed but the controlling variable is the propeller pitch (r.h.s. of Fig.3) was called "pitch control of shaft speed" in *Grimmelius and Stapersma (2003)* and was already presented in *Grimmelius and Stapersma (2000)* as "pure pitch control", the latter of course sinning against the preferred terminology introduced above and "shaft speed control by pitch" would be the better name. Now the input demand system can define any combination of fuel demand and pitch demand, so again a "combinator curve" is possible. For a given command the fuel rack is steady, but of course in seaway all disturbances of shaft speed must be counteracted by

frequent movements of the pitch that at present are not allowed by Class rules.

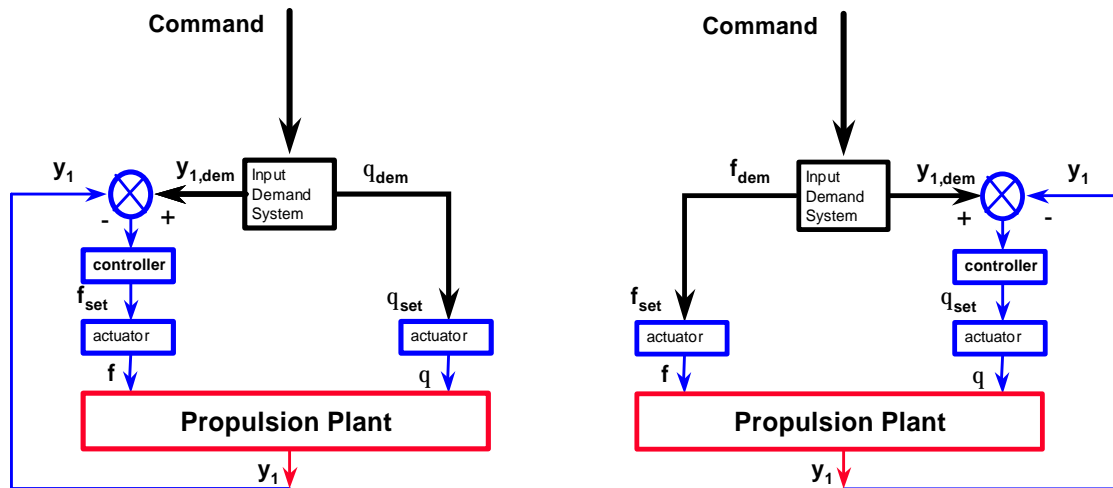
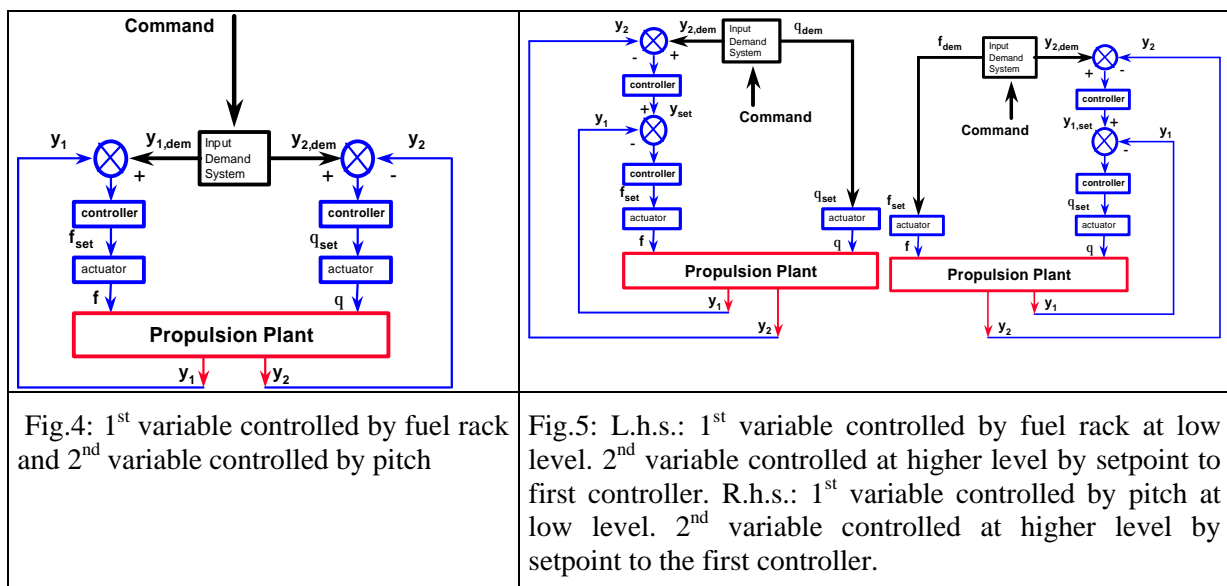


Fig.3: Single output control (e.g. shaft speed control if $y_1 = n$) by fuel rack (left) or pitch (right)

2.4 Control of two outputs

If in the previous Fig.3 the controlled variable is not shaft speed but e.g. ship speed, torque, thrust or any other variable that might be interesting to control, one might still want to keep an eye on shaft speed. In that case the first controlled variable (y_1) still is shaft speed which can classically be controlled by changing the fuel rack in a governor, while the second controlled variable (y_2) can be controlled by another controller that acts on the pitch, Fig.4. Both controllers work in parallel and act with their own input on the system. But of course it would be possible to change sides and give speed control to the pitch controller and the other controlled variable to the fuel rack controller. But it is not necessary to include shaft speed as controlled variable and in fact any two variables may be selected. But then also a choice must be made which one is controlling through the fuel rack controller and which one through the pitch controller, at least when one does not combine the controller into a multiple input - multiple output (MIMO) controller.



As an alternative the second controlled variable (y_2) can again be controlled by another controller but now one that provides a set point to the first controller (e.g. the engine governor), refer to the left hand side of Fig.5. Now both controllers are in series and it remains to be seen whether that will result in a stable system. The propeller pitch, following a demand from the input demand system will exhibit

quiet behaviour, also in seaway. As before, these two controllers could be shifted to the pitch side with y_1 as the controlled parameter directly driving the pitch controller, while y_2 is controlled at a higher level by a controller that provides a set point to the lower level pitch controller, Fig.5 r.h.s. Given any pair of controlled variables y_1 and y_2 , the schemes in Fig.4 and Fig.5 give two times three, is six, different control strategies.

3. Control objectives

3.1 Overview of candidates for control

The controlling variables were:

- * Fuel rack
- * Propeller pitch

The following candidates are considered as the controlled variables:

- * Shaft speed n
- + Ship speed v_s
- + Thrust T
- * Torque Q
- Torque coefficient K_T or C_T
- Thrust coefficient K_Q
- (Advance coefficient J)
- Exhaust temperature T_{exh}
- Thermal wear parameter diesel engine
- Mechanical wear parameter diesel engine
- Mechanical wear parameter pitch control system

These are not equivalent variables as discussed below. The meaning of + and * in the listing will be explained below. Before contemplating what to control under dynamic conditions such as ship acceleration and sea-state, we have to consider the steady state requirements of control.

3.2 Steady state control

Ship speed is in a way the ultimate objective of the "mobility" function of the ship. Any variable that in steady state conditions in some way is proportional (not necessarily linear though) with ship speed is a variable with which it is possible to *command* the speed of the ship, i.e. the variable is said to be a "commanding variable". These variables are shown with a + in the list above. Obviously ship speed itself is "commanding" and also all variables that are somehow proportional to the energy input to the system, i.e. thrust and torque and also fuel rack (but this is a controlling input and not a controllable output). Torque and fuel rack under certain conditions are not commanding and therefore they are given an asterisk (*) instead of a + in the list.

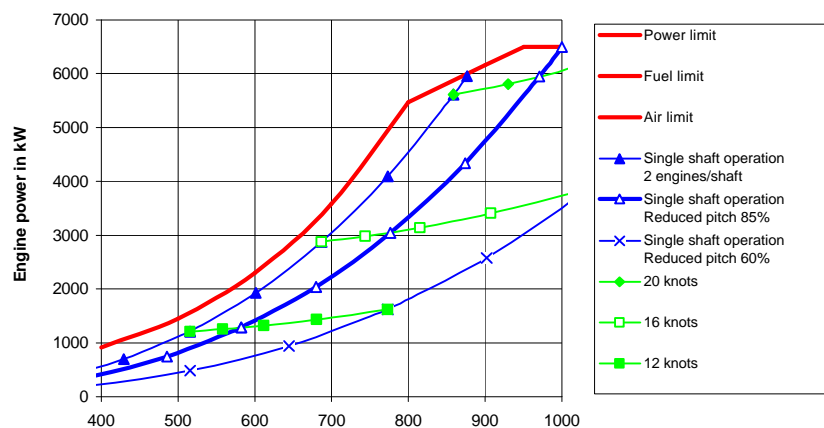


Fig.6: Engine power/speed envelop with lines of constant pitch and lines of constant ship speed

Shaft speed, strange as it may seem, in essence is not "commanding" by itself since pitch variation makes that a certain shaft speed can be associated with any ship speed. And pitch as such also is not

"commanding" by itself since a variation of shaft speed makes that a certain pitch can also be associated with any ship speed. This is shown in Fig.6 taken from *Klein Woud and Stapersma (1993)*. Combinations of the two however are "commanding" and these are the well-known combinator curves. Limit cases for these combinator curves are:

- constant pitch, in which case shaft speed is a commanding variable (the normal fixed pitch propeller (FPP) case)
- constant speed, in which case the pitch is a commanding variable (the ultimate CPP case)

Shaft speed and pitch therefore are marked with an asterisk (*) in the list above.

Starting with, e.g., shaft speed and pitch in the case of shaft speed control by fuel rack, the input demand signals for fuel rack, thrust, torque etc. can be calculated by running a simulation until steady state is reached. In order to do so a certain combinator curve was assumed, Fig.7. In order to cover all possible cases the combinator curve has a constant (low) shaft speed for low ship speeds, a constant (but not maximum) pitch regime for intermediate ship speeds, and a constant shaft speed again at the higher end where maximum ship speed is attained by increasing the pitch to maximum.

The choice for a combinator curve containing the two limit cases "constant shaft speed" and "constant ship speed" was deliberate (a) because it is the basis for control schemes on board of naval ships and (b) because it will prove to uncover the control strategies that are not able to command ship speed. Note that both torque and fuel rack are very flat at low ship speed so in this region these two variables are hardly able to command the ship. Also J , K_T and K_Q are almost constant during constant pitch, as follows from elementary considerations as given e.g. in *Klein Woud and Stapersma (1993)*. The last stage of pitch increase was inserted to provide some margin for over-torque. Also, since both speed control and pitch control are not against their limits, it offers the possibility to investigate seaway and the effectiveness of disturbance rejection later on. The alternative of a combinator curve where shaft speed and pitch vary continuously, as is often the practice for merchant ships, has not been chosen.

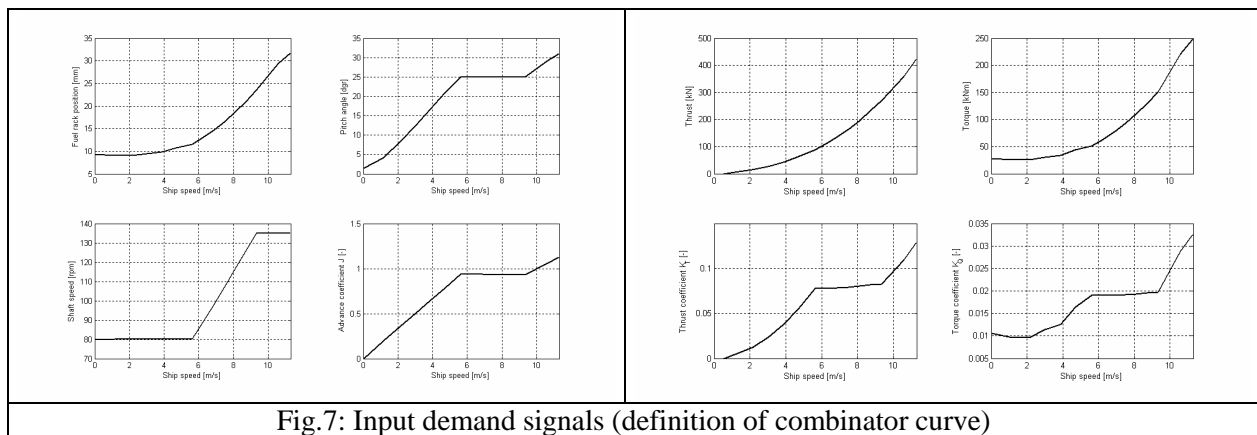


Fig.7: Input demand signals (definition of combinator curve)

3.3 Candidates for dynamic feedback control

Both ship and shaft speed, from a control theory point of view, are state variables since they are outputs of integrating processes, Fig.1. Thrust is input to one of these integrators and as such directly causing the ship speed. From an energy point of view torque times shaft speed (delivered power) is causing thrust times ship speed (effective power), although pitch variation influences the propulsion efficiency somewhat. Torque, like thrust, is proportional to ship speed (not linear of course) and thus "commanding", apart from the low speed range as was already observed. Torque is easier to measure than thrust and in fact is already measured on most naval ships, but in the merchant marine this not yet common practice.

The other variables are not "commanding" or at least are not meant to be. In fact they specify the conditions under which the ship speed is reached or maintained. This certainly applies to the non-dimensional variables associated with the propeller operating point. Optimising propeller performance

means controlling such a propeller "internal" parameter. At first the advance coefficient or the angle of the flow at some distance of the blade seems logical (D is the propeller diameter):

$$J = \frac{v_A}{n \cdot D} \quad \beta = \arctg\left(\frac{v_A}{0.7 \cdot \pi \cdot n \cdot D}\right) = \arctg\left(\frac{J}{0.7 \cdot \pi}\right)$$

But these are not a measure of the angle of attack and that for two reasons. First for that purpose the pitch angle p must be subtracted and secondly it is the variations of the advance velocity v_a that are causing the all important load variation of the propeller blade. Now although these wake speed variations can, given the sea state, be predicted nowadays with advanced software, it seems far away to measure these speeds continuously on board for the purpose of control. But the effect of angle of attack (and thus initiation of cavitation) perhaps can be deduced from a torque or thrust coefficient. Several definitions are possible:

$$K_T = \frac{T}{\rho \cdot n^2 \cdot D^4} \quad K_Q = \frac{Q}{\rho \cdot n^2 \cdot D^5} \quad C_T = \frac{T}{\frac{1}{2} \cdot \rho \cdot v_A^2 \cdot \frac{\pi}{4} \cdot D^2} \quad C_T = \frac{8}{\pi} \cdot \frac{K_T}{J^2}$$

Four quadrant measurements often are presented as:

$$C_T^* = \frac{T}{\frac{1}{2} \cdot \rho \cdot \{v_A^2 + (0.7 \cdot \pi \cdot n \cdot D)^2\} \cdot \frac{\pi}{4} \cdot D^2} \quad C_Q^* = \frac{Q}{\frac{1}{2} \cdot \rho \cdot \{v_A^2 + (0.7 \cdot \pi \cdot n \cdot D)^2\} \cdot \frac{\pi}{4} \cdot D^3}$$

But these are again dependent on the basic variables:

$$C_T^* = \frac{8}{\pi} \cdot \frac{K_T}{J^2 + (0.7 \cdot \pi)^2} \quad C_Q^* = \frac{8}{\pi} \cdot \frac{K_Q}{J^2 + (0.7 \cdot \pi)^2}$$

ρ is the seawater density. A (part of) the lift force of the blade is related to (part of) the entrance velocity head of the water. Having either T or Q (both resulting from the lift force on the blade) in a controlled variable means that the information of the angle of attack of the flow enters the equation. C_T does not seem practical, since both thrust and advance velocity are at least difficult to measure. The same applies to C_T^* . C_Q^* containing advance velocity is not practical either. This leaves K_T and K_Q . There is a preference for K_Q since both torque and shaft speed can easily be measured.

The exhaust temperature and other engine internal variables of the engine as well as any associated wear limits are part of a wider research, see *Grimmelius and Stapersma (2000)*, *Grimmelius and Stapersma (2001)* and *Grimmelius and Stapersma (2003)*. This also applies to internal variables of the propeller pitch mechanism and its associated wear parameters. The present paper will confine itself to the core propulsion system and not go into the details of either the engine or the pitch actuating mechanism and therefore the model used in this limited research has no sophisticated sub-models for these system parts. The engine e.g. is represented by its steady state power - speed map and the pitch mechanism is just a first order system following the setpoint.

Then the list of controlled variables could be as shown in Table 1. Feed forward control through fuel rack and pitch ("no control") as well as feed back control of shaft speed either by controlling the fuel rack or the pitch will be considered as basic cases. Single control of ship speed, torque or thrust could be options or at least primitive cases for real options. Single control of torque or thrust coefficient will give a first indication of what happens when these variable are brought into the picture. Conservative or prudent behaviour would dictate that, when the option of two controlled variables is considered, shaft speed is the 1st controlled variable and either ship speed, thrust, torque, thrust coefficient or torque coefficient would be the 2nd controlled variable. But one could even, at least for the moment, disregard shaft speed and later fix that by additional half-sided control, i.e. control that becomes active only when minimum or maximum speed is reached, as already proposed in *Faber (1993)*. In that case controlling

- ship speed together with thrust or torque coefficient
- thrust together with thrust coefficient
- torque together with torque coefficient

could be viable options.

The numbers shown in the 3rd row of Table 1 are the number of control strategies that could be adopted according to Fig.2 to Fig.5. The last row designates which schemes have been looked into.

Table 1: Combinations of variables that present itself for control

y_1	-	n	v_s	T	Q	K_T	K_Q	n	N	n	n	n	v_s	v_s	T	Q
y_2	-	-	-	-	-	-	-	v_s	T	Q	K_T	K_Q	K_T	K_Q	K_T	K_Q
<i>Nr of strategies</i>	1	2	2	2	2	2	2	6	6	6	6	6	6	6	6	6
<i>Investigated</i>	1*	2*	2			2	2				6	+		6		
<i>"Commandable"</i>	1	2	2			1	1				4?			3?		
<i>Presented</i>	+	+				+	+				+					

4. Cases

4.1 Basic Cases: no feedback control and single control of shaft speed

The basic cases are "no feedback control" or rather "feed forward control of fuel rack and pitch" according to Fig.2 and feed back control of shaft speed by either fuel rack or pitch basically according to Fig.3 and actualised in Fig.8. The input demand signals required for these cases are fuel rack, pitch and shaft speed and were derived with the help of Fig.7. The results of command tracking are presented in Fig.9, Fig.10 and Fig.11 for an acceleration in the lower speed regime (constant shaft speed) followed by a further acceleration in the intermediate speed regime (constant pitch).

Controlling shaft speed by fuel rack (governor control) has the potential of the fastest acceleration albeit at the cost of over-fuelling the engine which for a real case will be limited one way or the other. The effect of this can be seen in the "no control" case where fuel is limited to the new setpoint value, resulting in a dip in the shaft speed during the 1st part of the acceleration but during the 2nd part of the acceleration a trajectory in the engine map that almost avoids the overload area. The excursions of J , K_T and K_Q are quite significant for governor control and no control, although slightly less for the latter due to the limited step in fuel rack.

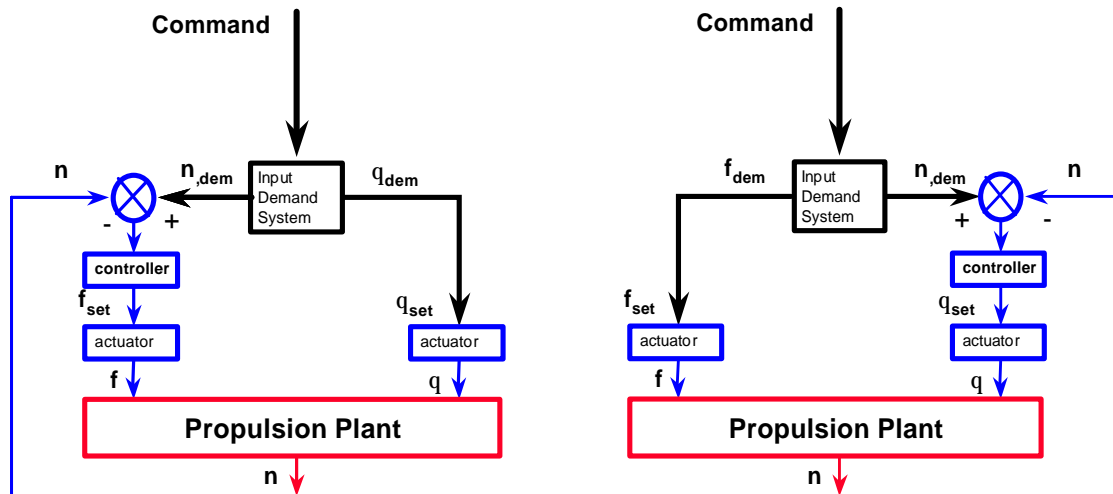


Fig.8. Shaft speed control by fuel rack (left) or pitch (right)

Same as the behaviour of the engine can be judged from the trajectory in the engine map, so the "well being" of the propeller can be deduced from the trajectory in a $K_T - J$ or $K_Q - J$ diagram. In these propeller maps the fast change to the high pitch value is illustrated as well as the fact that thrust and torque coefficient are then bounded by that constant pitch. The rapid change of pitch together with the inclination of K_T and K_Q cause the high peak values of the latter, but also the high thrust from which

stems the excellent acceleration of the ship.

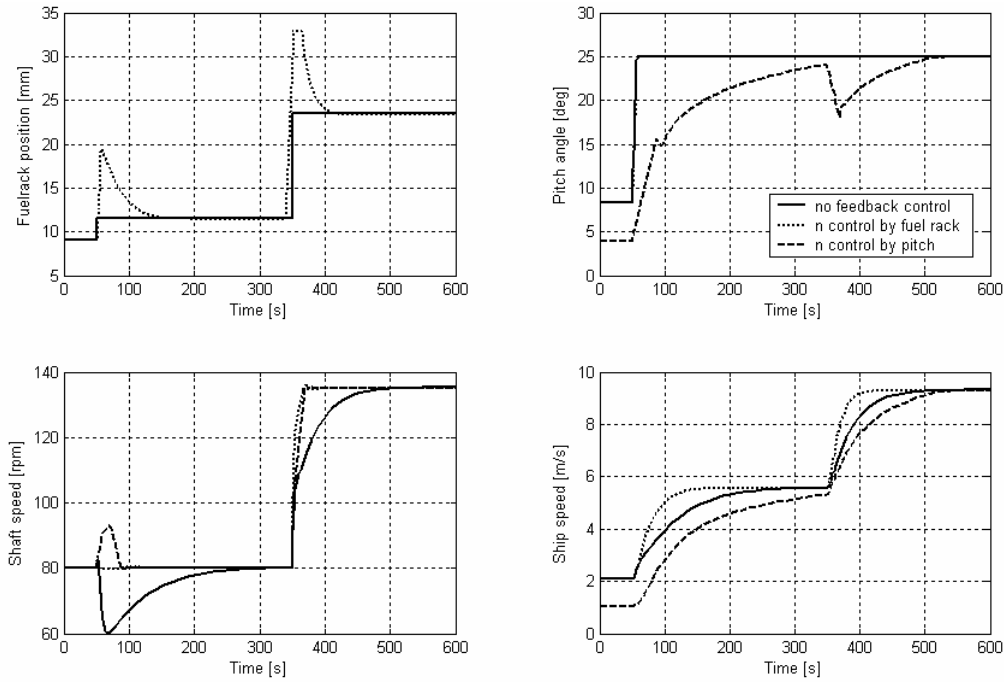


Fig.9: Time tracks of fuel rack, pitch, shaft speed and ship speed for basic cases

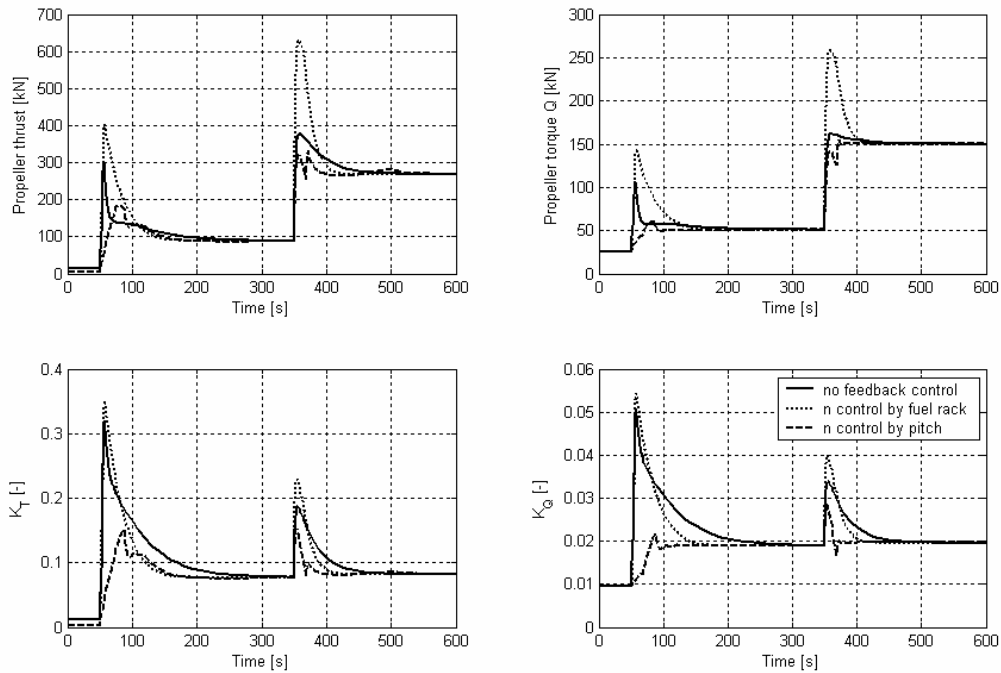


Fig.10: Time tracks of propeller thrust, torque, thrust coefficient and torque coefficient for basic cases

Controlling shaft speed by pitch has the problem that the fuel demand, directly coming from the input demand system is so flat that it is not well capable of commanding the ship speed: it has a considerable lower initial ship speed than the other two control strategies. During the 1st part of the acceleration the pitch increases slowly, in fact it is too slow to absorb the power that comes available from the small step change in fuel rack. Due to the flat fuel rack characteristic at low ship speed, all extra fuel is available for shaft acceleration and as a result shaft speed increases before the pitch is able to keep it at the fixed minimum value again. During the 2nd part of the acceleration the speed error reduces the pitch for more than a minute. This tends to slow down the acceleration. But although J dips considerably the pitch reduction also reduces the excursions of K_T and K_Q indicating that the

angle of attack of the water flow over the blade remains closer to its design value.

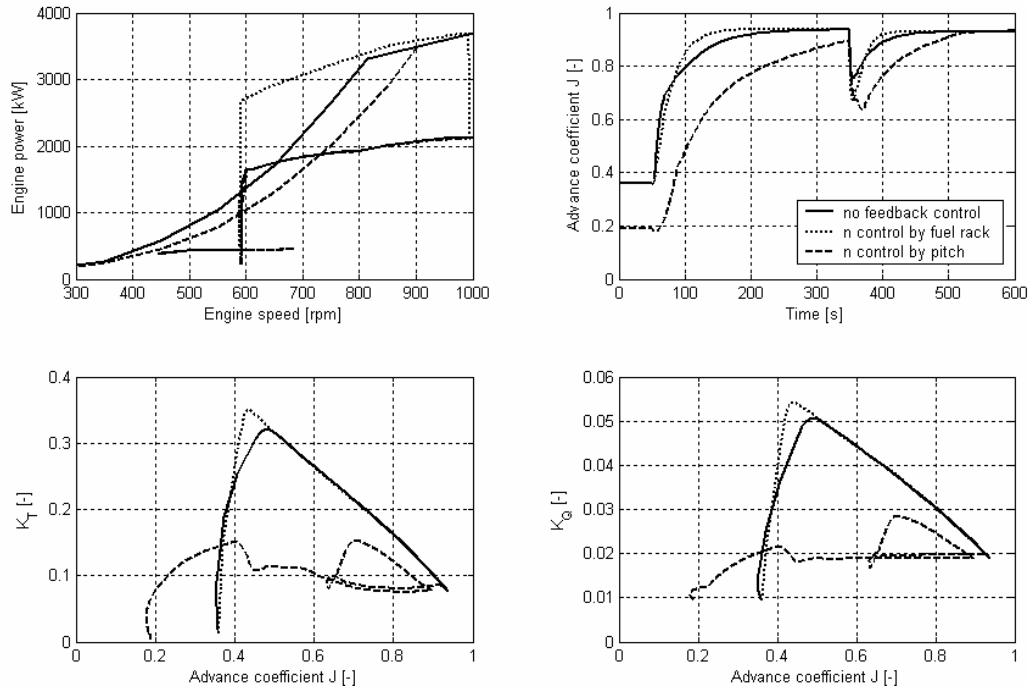


Fig.11: Trajectories in engine power/speed map, time track of advance coefficient J and trajectories in K_T - J and K_Q - J diagram of propeller for basic cases

4.2 Single control of thrust and torque coefficient

Fig.7 shows the associated input demands for K_T and K_Q . For the intermediate ship speeds, pitch, K_T and K_Q all are constant. Thus controlling either thrust coefficient or torque coefficient by fuel rack ($y_I = K_T$ or K_Q in the l.h.s. of Fig.3) leaves the input demand system with two constant demand signals during the intermediate speed regime (where pitch is constant and thus not "commanding"). This makes the system unable to command the ship speed in that speed regime. So K_T or K_Q can only be controlled by controlling the pitch (l.h.s. of the basic scheme Fig.3 and actualised in Fig.12), since that leaves the fuel rack on the other side as "commanding" input to the system.

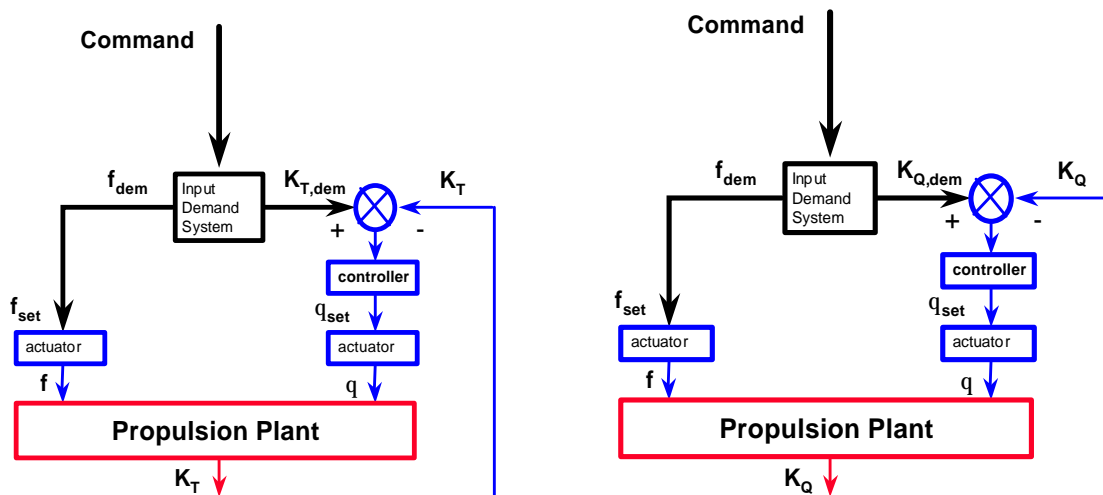


Fig.12: Single control of K_T (left) or K_Q (right) by pitch. Control of K_T and K_Q by fuel rack cannot command the ship speed if pitch is constant on the combinator curve and are therefore not shown

The same two accelerations were carried out as before, but now with single control of thrust and torque coefficient by controlling the pitch, while giving fuel rack the demanded setpoint from the

combinator curve. The results are shown in Fig.13, Fig.14 and Fig.15.

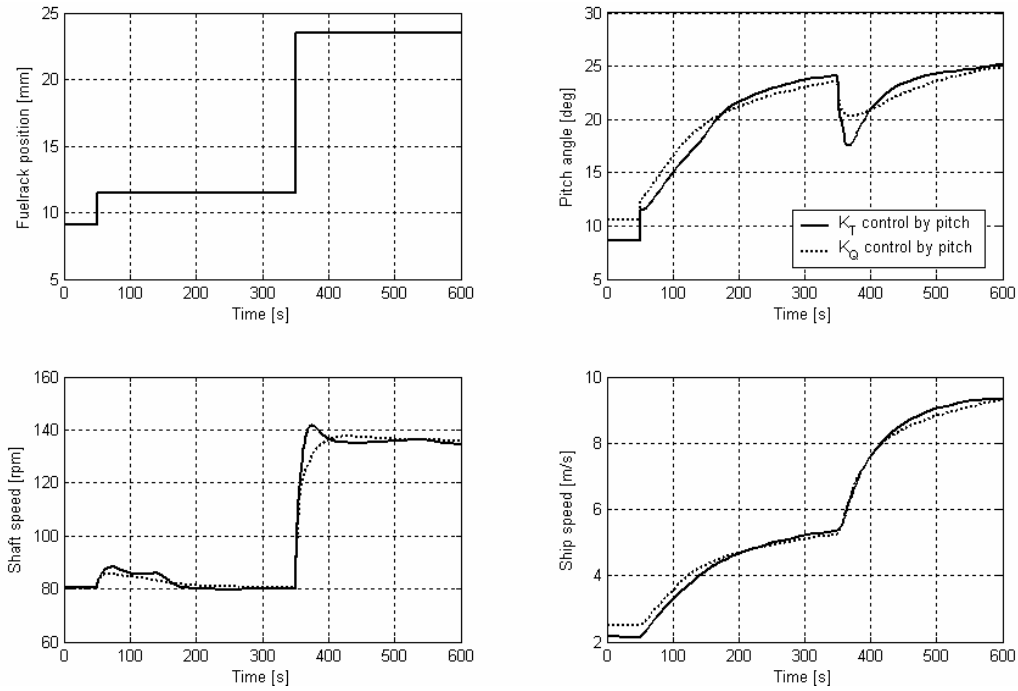


Fig.13: Time tracks of fuel rack, pitch, shaft speed, ship speed for single control of K_T or K_T by pitch

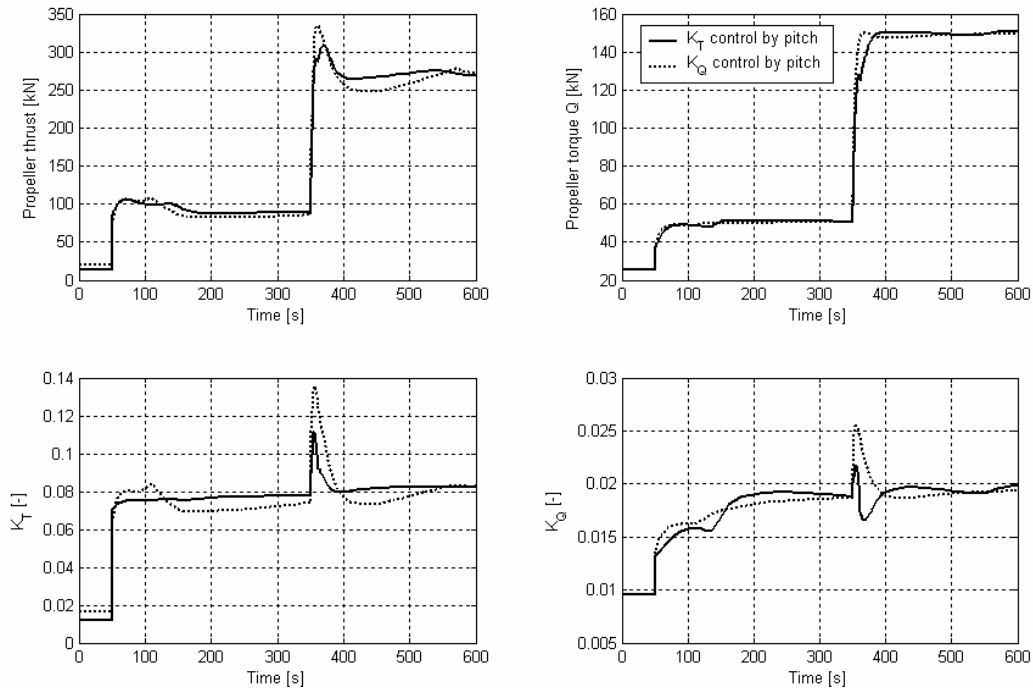


Fig.14: Time tracks of propeller thrust, torque, thrust coefficient and torque coefficient for single control of K_T or K_Q by pitch

Again the steady state initial low speed for the two cases is not the same, the K_Q -control being at fault. This is due to the fact that K_Q is almost constant in this area and that its setpoint together with the almost flat setpoint of the fuel rack is not capable of commanding ship speed. Shaft speed, although not controlled explicitly here, still remains pretty constant in both cases. The initial step in fuel rack causes the shaft speed to rise. The flat steady state value of the fuel rack indicated that this is the only way to get rid of the extra energy at these low speeds. When at 4 kn (150 sec) the steady state fuel rack and thus the demanded energy flow of the system begins to increase, the acceleration is quasi-

stationary, i.e. all variables remain close to the steady state values of Fig.7. During the 2nd part of the acceleration the shaft speed goes up and tends to overshoot. But control of thrust or torque coefficient is capable of sensing that quickly and the subsequent reduction of pitch keeps the shaft speed within close limits. Thrust coefficient seems slightly more effective in this respect. Also K_T control seems more effective in reducing the excursions of both K_T and K_Q . The trajectories in the propeller maps and in the engine map also indicate a "mellow" behaviour. But overall acceleration of course is slow and in fact comparable to shaft speed control by pitch as shown previously.

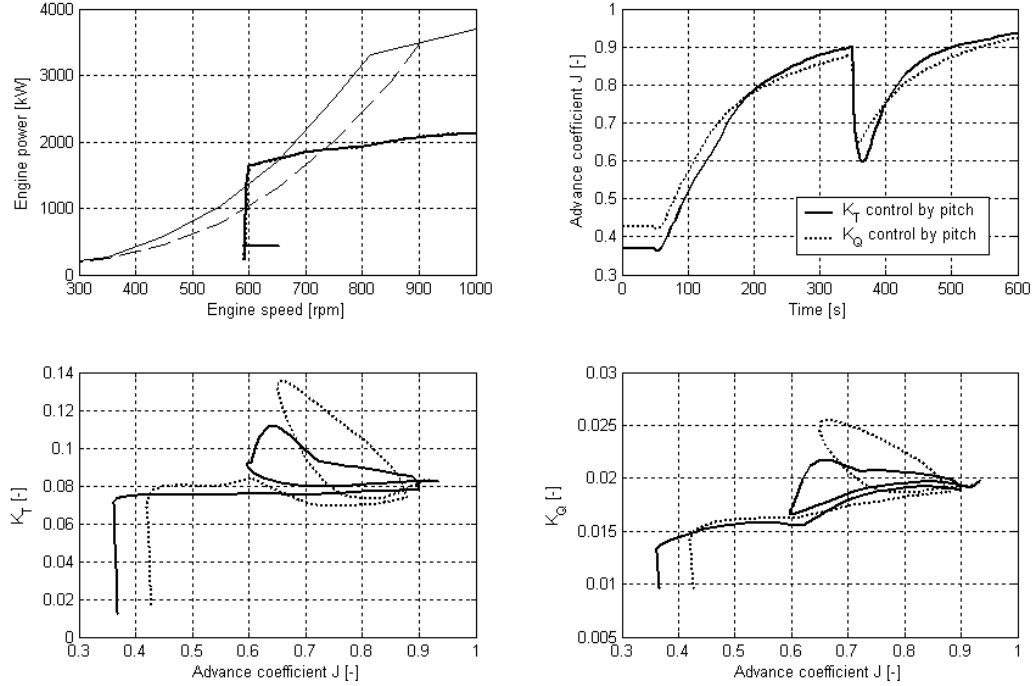


Fig.15: Trajectories in engine power/speed map, time track of J and trajectories in K_T - J and K_Q - J diagram of propeller for single control of K_T or K_T by pitch

4.3 Combined control of thrust coefficient and shaft speed

Considering control of thrust coefficient the more effective it will now be combined with shaft speed control. According to our previous analysis there are 6 possibilities, as depicted in principle in Fig.4 and Fig.5. The control "over two sides" of Fig.4 is actualised into the two cases shown in Fig.16 and the corresponding results for the two-stage acceleration in Fig.17, Fig.18 and Fig.19.

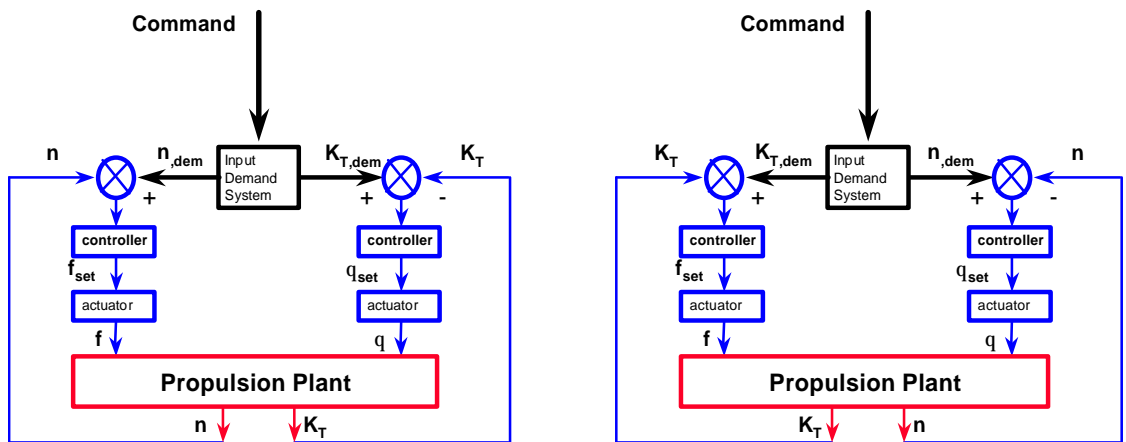


Fig.16: Shaft speed control by fuel rack, K_T control by pitch (left) and K_T control by fuel rack, shaft speed control by pitch (right)

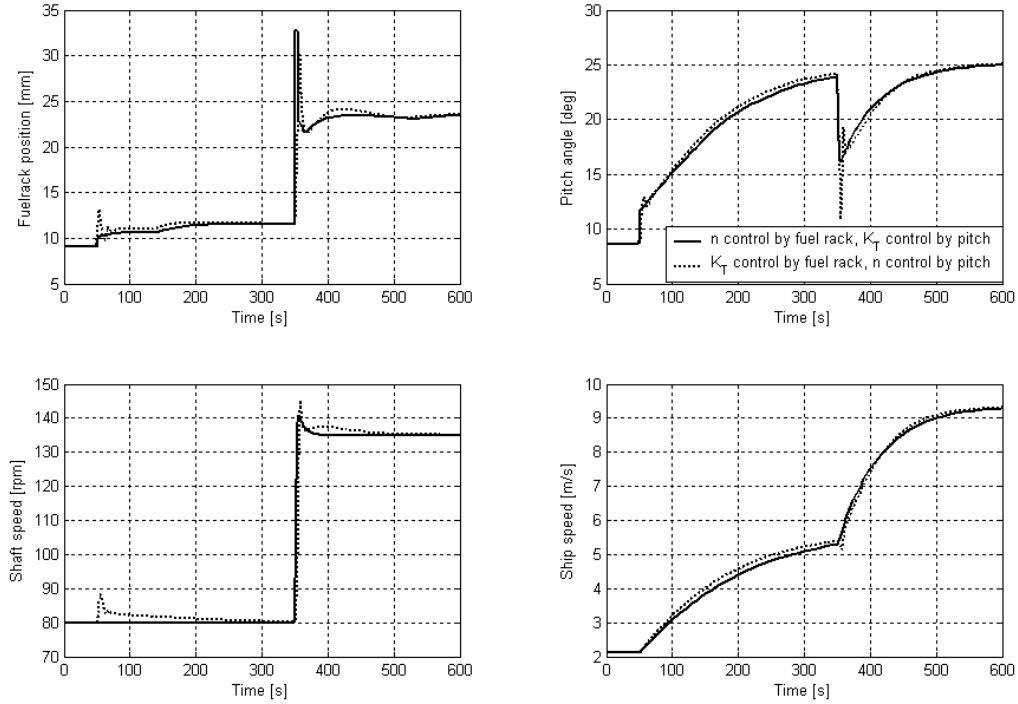


Fig.17: Time tracks of fuel rack, pitch, shaft speed and ship speed for control of n and K_T

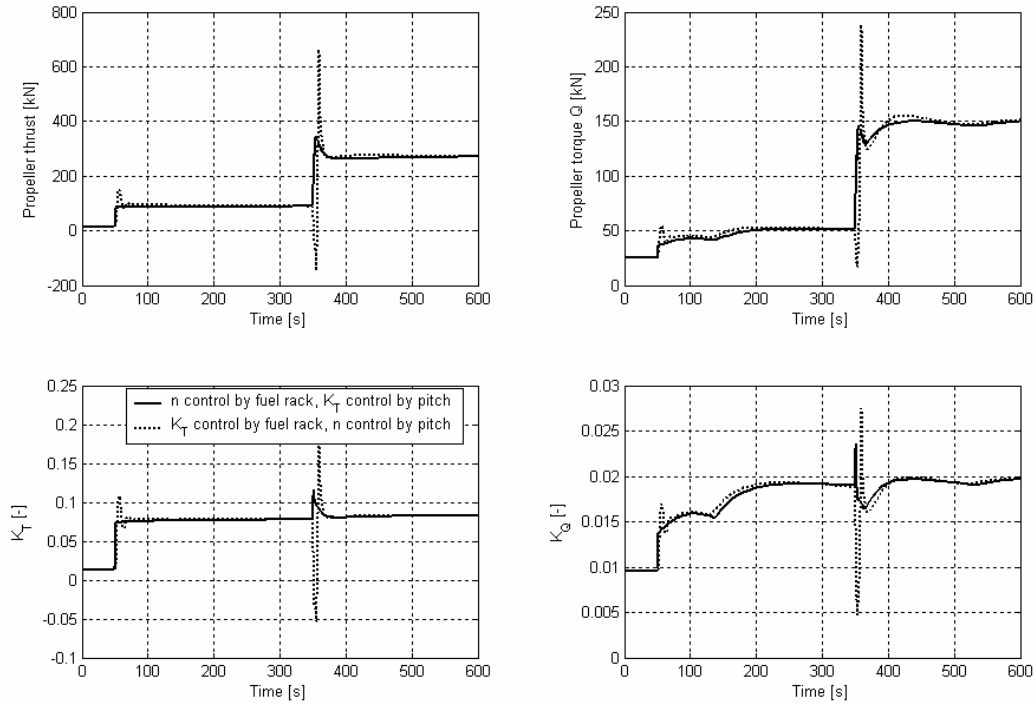


Fig.18: Time tracks of thrust, torque, thrust coefficient, torque coefficient for control of n and K_T

The control of K_T by fuel rack and shaft speed by pitch (Fig.16 r.h.s., dotted line in subsequent figures) does seem to be stable. The pitch moves wildly during the begin of the first accelerations, while shaft speed still overshoots. At the start the 2nd part of the acceleration the pitch is also retracted fiercely but effectively since the trajectory in the engine map remains under the overload line. The other way round, i.e. control of shaft speed by fuel rack and K_T by pitch (Fig.16 l.h.s. and straight line in subsequent figures) seems more logical. During the first part of the acceleration shaft speed surely is kept constant, and also thrust coefficient is controlled very well. At the start of the 2nd part of the acceleration shaft speed exhibits a slight overshoot although fuel rack goes up fast. The trajectory in

the engine map runs through the overload area, almost the same as the basic case with single shaft speed control of the fuel rack. But pitch reduction on the other side makes that K_T remains pretty constant and its trajectory in the propeller diagram is offered to hydrodynamic experts for inspection.

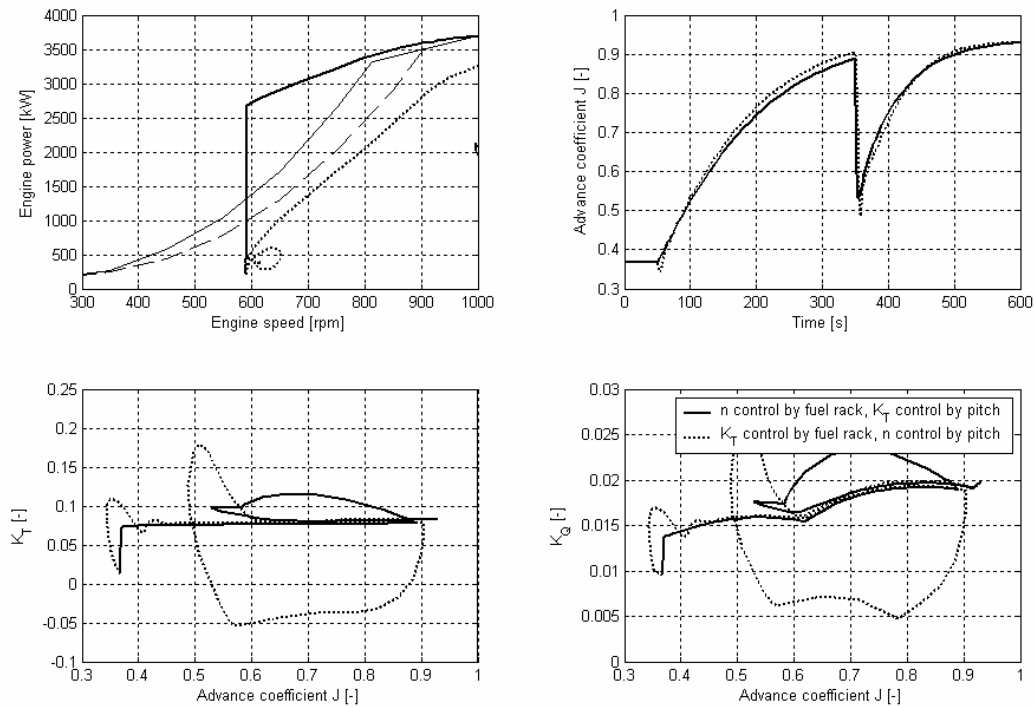


Fig.19: Trajectories in engine power/speed map, time track of J and trajectories in K_T - J and K_Q - J diagram of propeller for control of n and K_T

The four other cases, based on Fig.5, were investigated but the results are not shown here. Control over the fuel side only, i.e. the left hand side of Fig.5, can be accomplished in two ways. When shaft speed is controlled at the higher level and produces a setpoint for K_T control by fuel rack at a lower level, the response is very much the same as ordinary single shaft speed control. The excursions of K_T and K_Q are almost as severe. The reason is that the pitch is allowed to move to its new setpoint quickly, limiting any possibility to restrict the propeller coefficients. When K_T is controlled at the higher level and produces a setpoint to speed control by fuel rack (i.e. a normal governor) at a lower level, we have the same problem as seen before: ship speed cannot be commanded since the setpoints of K_T and pitch remain constant during parts of the combinator curves where pitch is constant.

Control over the pitch side alone, i.e. the r.h.s. of Fig.5 also can be accomplished in two ways. When shaft speed is controlled at the higher level and produces a setpoint for the K_T control by pitch at lower level, a very unstable behaviour of pitch was observed causing also excursions of K_T and overshoot of shaft speed. The other way round, i.e. control of K_T at the higher level producing a setpoint for shaft speed control by pitch at the lower level produces a behaviour almost the same as K_T control alone: shaft speed has the same (small) overshoot indicating that in this case shaft speed control is hardly effective. Also the ship speed cannot be commanded properly at low speeds.

All cases of "control over one side" are examples of multiple, here two, input and single output control (MISO control). The selection of one of the variables as "high level control" is just one way of assigning weights to the two input variables in a MISO control strategy.

4.4 Other cases

A systematic scan of all other combinations of variables that can be controlled, Table 1, is in progress. Ship speed control has been investigated but seems not so useful apart from a possible combination

with either K_Q or, with the experience already gained, rather with K_T . Although control of thrust T together with thrust coefficient K_T looks interesting, the results of just K_T control show, Fig.14, that thrust is already constant for that case, as was shaft speed. So there is a good reason to use the fuel rack for something else, e.g. limiting some engine parameter. Fig.15 shows that the engine still runs into the overload area. This will be part of future work. Another topic not yet investigated thoroughly is disturbance rejection, i.e. the effect of waves and the dynamic loading of the blades.

5. Conclusions

Classical propulsion control where shaft speed is controlled by fuel rack and pitch is given a feed forward schedule by the input demand system results in fast accelerations, but tends to overload the engine. This of course is well known and normally several schemes are adopted to either limit the fuel rack or limit the pitch rate. Limiting the fuel rack sometimes is speed dependent but then the danger exists that the engine will decelerate and ultimately will be stopped. A more sophisticated way is a charge pressure dependent limiter as is found on almost every governor (although it is not always used!). It is even better to limit the pitch rate of change, but when the pitch is already at its setpoint (as in the 2nd part of our accelerations) this can only work if the pitch is retracted somewhat (as the French say: "reculer pour mieux sauter").

But when considering the propeller behaviour the thrust coefficient presents itself as a logical variable that could be controlled. This thrust coefficient is best controlled through the pitch rather than through the fuel rack since in the latter case the control system cannot have command of ship speed for constant speed parts of a combinator curve. The resulting control scheme, i.e. K_T control by pitch shows an almost ideal behaviour: pitch is changed gradually and if it is already at its setpoint it is retracted somewhat and then slowly comes back to its setpoint. It is expected that this way cavitation can be suppressed. Ship acceleration however is slow and this could be unwanted in certain circumstances.

A refinement however is to allow higher values of K_T than the value corresponding to the new setpoint. In fact one could adopt two levels, one to just allow cavitation-free acceleration, the other being optimised for acceleration when manoeuvring in narrow waters.

It must be stressed that no thorough optimisation of the control parameters was carried out, so the results may still improve and are only meant as a guidance to a first selection.

The proposed control scheme diverges from current practice in many ways:

- Thrust has to be measured which certainly is not yet common practice. An alternative could be to measure torque and use knowledge of the propeller characteristic to estimate thrust.
- Shaft speed is measured and used for control, but not via a classical speed governor that controls the fuel rack. Shaft speed however is effectively controlled through K_T and only near maximum rpm of the engine additional measures need be taken.
- Pitch control must be possible. This would require a conversion of attitude of classification societies to this matter.
- Fuel rack is available for additional limiter functions primarily intended to prevent the diesel engine from overspeed and thermal overload.

Using the fuel rack for shaft speed control next to K_T control by pitch did not result in better results and although it is understandable to cling, at least in part, to the trusted governor, it seems that the scheme as proposed is safer overall. Further systematic research of other combinations however is required to ensure that no promising combinations are forgotten.

Acknowledgement

The authors want to take the opportunity to thank Senter for their support of the project.

References

- GRIMMELIUS, H.T.; STAPERSMA, D. (2000), *Control optimisation and load prediction for marine diesel engines using a mean value simulation model*, ENSUS 2000, Newcastle-upon-Tyne
- GRIMMELIUS, H.T.; STAPERSMA, D. (2001), *The impact of propulsion plant control on diesel engine thermal loading*, 23rd CIMAC Conf., Hamburg
- SPRONSEN, P.J. VAN; TOUSSAIN R.L. (2001), *A control solution to marine diesel engine overloading aboard karel doorman class frigates*, CAMS, Glasgow
- GRIMMELIUS, H.T.; STAPERSMA, D. (2003), *Analysis of the impact of control strategy on internal component loading for a ship propulsion plant*, 13th Ship Control Systems Symp., Orlando
- SCHULTEN, P.J.M.; TOXOPEUS, S.L.; STAPERSMA, D. (2004), *Propeller-diesel engine interaction in a turn*, 7th INEC Conf., Amsterdam
- SKOGESTAD, S.; POSTLETHWAIT, I. (1996), *Multivariable Feedback Control*, Wiley, New York
- FABER, E. (1993), *Some thoughts on Diesel Marine Engineering*, SNAME Transactions, Vol. 101
- KLEIN WOUDE, J.; STAPERSMA, D. (2002), *Design of propulsion and electric power generation systems*, ISBN 1-902536-47-9, IMarEST, London

Distributing Operational Simulation Systems across South African Ports

Desmond Simpson, National Port Authority, Johannesburg/South Africa, dess@npa.co.za

Ludwig Furstenberg, ITE Consulting, Durban/South Africa, ludwigf@ite.co.za

Abstract

Virtual Prototyping is finding an increasingly strong foothold in the maritime industry as an efficient tool to scientifically evaluate multiple options with applications ranging from simulation of detailed process flow analysis or construction techniques in shipyards to long-term logistic port planning and development strategies. Heuristic based discrete event simulation models have been successfully developed and applied to assist the National Ports Authority of South Africa (NPA) with the development of port infrastructure to cater for future industrial expansions and cargo growth, de-bottlenecking process flows and numerous capacity studies during the past decade. The benefit of identifying potential problems of proposed expansions in a low cost environment has now been extended in a nationwide information system that allows users in any port in South Africa to configure a simulation run and receive results via an automated simulation engine called The Big Picture System.

1 What is Simulation/Virtual Prototyping?

Simulation refers to a broad collection of methods and applications to mimic the behaviour of real systems, usually on a computer with appropriate software. Today, simulation is more popular and powerful than ever since computers and software are more advanced.

People often study a system to measure its performance, improve its operations, or design it if it does not exist. There has even been instances where managers requested that a simulation be developed but did not really care about the final results. Their primary goal was to focus attention on understanding how their system currently worked. Often simulation analysts find that the process of defining how the system works, which must be done before you can start developing the simulation model, provides great insight into what changes need to be made.

There are many different types of models. Maybe the first thing the word evokes is a physical replica or scale model of a system, sometimes called an *iconic* model.

A logical model is just a set of approximations and assumptions, both structural and quantitative, about the way the system works. A logical model is usually represented in a computer program that's exercised to address questions about the model's behaviour; if the model is a valid representation of the real life system, one expects to learn about the system's behaviour too.

If the model is simple enough, traditional mathematical tools like queuing theory, differential equations, or something like linear programming might suffice to arrive at the answers. However, most systems that people model and study are pretty complicated, so that valid models of them are pretty complicated too. A simple model of a complicated system can always be developed, but it may not be valid. Analysing such a model may yield nice, simple answers to the wrong questions!

Over the last two or three decades, simulation has been consistently reported as the most popular operations research tool, *Kelton et al. (2002)*. The main reason for simulation's popularity is its ability to deal with very complicated models of correspondingly complicated systems. These makes it a versatile and powerful tool. Another reason for simulation's increasing popularity is the obvious improvement in performance/price ratios of computer hardware.

The Bad News is that simulation is not a paradise. Many real systems are affected by uncontrollable and random inputs, many simulation models involve random, or stochastic, input components, causing

their output to be random too. Running a stochastic simulation once is like observing one day of port operations – you will probably see something different next time. In many simulations, as the time frame becomes longer most results averaged over the run will tend to settle down and become less variable, but it can be hard to determine how long is “long enough”. The study might dictate that the simulation stop at a particular point (for instance, a canteen could be open from 9am to 5pm), so running it longer to calm the output is inappropriate.

Thus, you have to think carefully about designing and analyzing simulation experiments to take account of this uncertainty in the results, especially if the appropriate time frame for your model is relatively short.

2 What are Expert Systems?

In broad terms, Expert Systems are computer software packages that are able to apply expert reasoning to a particular problem or environment. Through a method of structured reasoning, the Expert System diagnoses a situation to reach a conclusion. The Expert System is programmed with the rules and procedures relating to how decisions should be made: this includes a variety of soft reasoning – “rules of thumb” and a great deal of process information, which is often difficult to express in conventional mathematical or statistical terms. A Knowledge Base is the name given to a set of Expert System reasoning: it is not just data or information, but encapsulates the knowledge of how to reason with the information.

More specifically, ITE-C has been using a commercial expert system shell from Gensym, called G2. This software is intended mainly for intelligent on-line control and applying expert reasoning to mimic the intelligent reactions of seasoned personnel in managing a dynamic system. The flexibility provided in this programming environment makes it possible to capture the idiosyncrasies contained in the dynamics of a real life system, by breaking down the decision blocks to practically any level required to solve a particular problem. Furthermore, the object orientation makes it possible to mimic not only real life items, such as berths, ships, cranes, conveyors, etc, but also abstract objects required for control, such as calling orders, maintenance schedules, berthing requests to name but a few. This representation of knowledge, rules and procedures enables a systems developer to capture and verify system operations with terminal operators.

3 NPA meet G2

ITE Consulting (ITE-C) and South Africa’s National Port Authority (NPA), have led the way in the International Port fraternity by implementing Gensym’s G2, a state of the art expert system shell, in the development, management and operation of South African ports. All seven of South Africa’s commercial ports are under NPA’s control. In 1992 the first expert system application was built to assist in distributing some 12 million tons of maize donated by foreign aid to 4 drought stricken Southern African states. Consequently ITE has been involved in developing simulation models for each of the South African ports and has conducted numerous capacity studies, planning expansions and assisting NPA to find non-capital solutions for capacity problems (optimising resources). ITE has also built and used expert system based simulation models in various process industries to perform scenario testing, risk analysis and design verification of various logistic systems.

Previous publications have included more detail on how to build a simulation model particular to the port environment, *Furstenberg (2000)*, *Simpson et al. (2003)*. It is the purpose of this article is to report on recent advances in South Africa that aims to distribute the simulation tools developed during the past number of years to the port planners throughout the country.

4 Purpose/Need for Big Picture

It has been NPA’s vision from the start to integrate all the simulation models into one central model. Client-server technologies have advanced to such a stage that it is now possible to deploy the models

across a wider user-base. That way, all the knowledge contained in these models can be retained, whilst at the same time giving more users access to the system; thus multiplying expertise across the entire enterprise.

The G2 environment supports full client-server functionality. However at a cost of roughly US\$75 000 per development license, there was a considerable financial hurdle to overcome. These models have been developed for strategic long term planning, thus do not require daily access. Consequently, multiple G2 licenses throughout the 7 commercial ports in South Africa will be highly under utilized.

The solution was to make the best use of a single user G2 license and distribute the required information transfer and interactions with the system through the much more cost effective Microsoft windows environment, as discussed next.

5 Solution – outline

The NPA Big Picture system was dreamt up to fulfill the requirements from NPA:

- Enable nationwide configuration of simulation runs, on any of the configured ports in the system
- Maintain an audible trace of the simulation performed on a central SQL database
- Allow users to edit and view information in the system
- Manage the execution of simulation requests.

Besides the major cost saving and getting increased return on the investment NPA made over the past decade, it also means that a decentralisation process has been started. Until recently, only a handful of recourses in NPA and ITE-C had access to these simulation models. There have been instances when port engineers were surprised to find that these tools have existed for so long without them being aware of the accelerated scenario-testing avenue that has already been established. At the very least, this whole process of distributing the simulation models throughout the ports in South Africa has lead to a higher awareness very useful tools that can be applied to assist in medium to short term strategic decisions.

The flip side of this initiative is that the port planners are effectively empowered to extend their skills to include the art of simulation, with the backing support of seasoned experts in this field. The avenue for this support is made possible by firstly having a system administrator (with years of experience in applying simulation studies) accept or reject configured simulation requests (with reasons) and secondly by having almost direct access to the system developers to clarify unexpected results, as discussed later in this paper.

Fig.1 shows a diagrammatic representation of the Big Picture system architecture. A SQL Server database contains all the information about previous runs and results files for all the models. This database resides on a server that is accessible from anywhere on the NPA wide area network (WAN) that connects all the commercial ports nationwide, as well as the head office in Johannesburg. For security and access control, no outside company can connect directly to the NPA WAN. However, a secure means of connectivity can be established through a Remote Access Service Server (RAS Server). RAS allows one PC to dial into another PC with a callback function with login and password control. This call-back function enables a further level of security, since it dials to a specific number, disallowing other connections to the system even if the login and password is circumvented

This RAS server does not only meet the network security requirements of NPA, but also enables a single user G2 license to be made available to multiple clients, hence enabling massive savings from a financial point of view. Let us explore this a little further.

All the simulation models for the ports that are part of the Big Picture system, resides on the G2 Server. With a single user G2 license, only one simulation model at a time can be loaded to perform

simulation runs. It typically takes between 20 minutes and 5 hours to perform a simulation run, depending on the particular model and level of detail simulated. It is expected that a typical year might see 100 – 150 simulation run request from all the ports in South Africa, thus one PC will be adequate to handle the workload by performing the simulation runs one at a time. There is some intelligence built into the RAS Server to allow queuing of multiple simulation runs, so the benefit of a major cost saving is off-set by some possible delay in getting the simulation results.

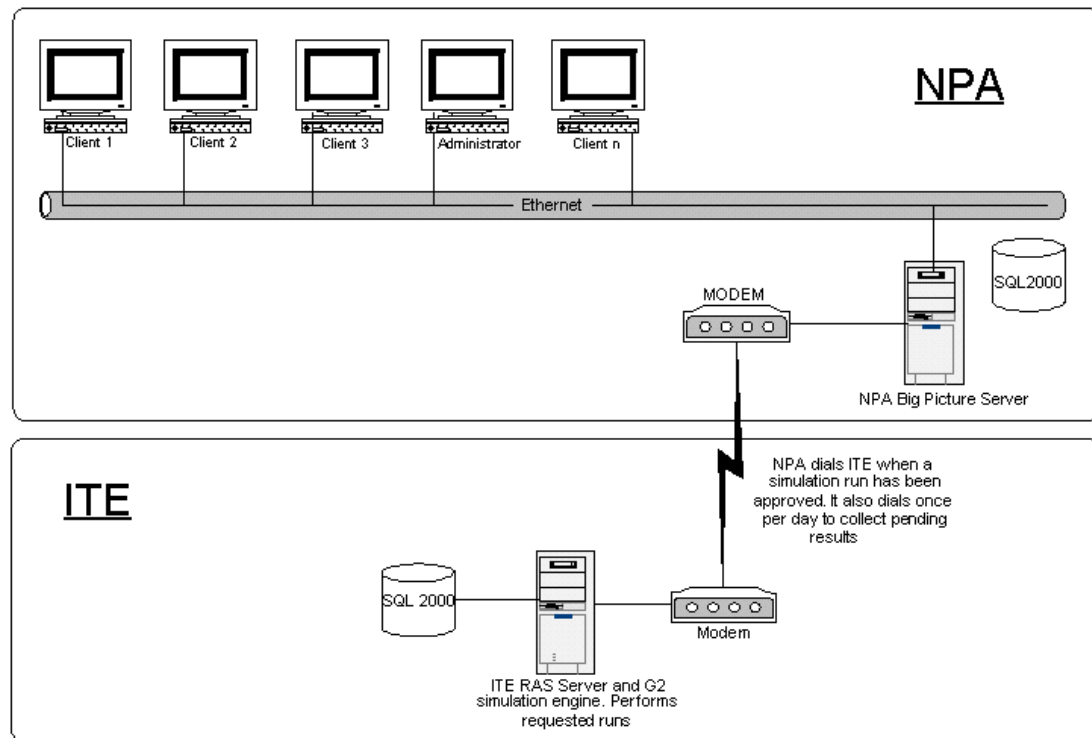


Fig.1: System architecture for the Big Picture system

The various G2 models thus contain all the operational rules and procedures for each specific port, while the SQL database contains the specific input information that configures or defines a specific simulation run (simulation experiment). It is thus only necessary to communicate the input information to the G2 Server and once the simulation run is completed, the result file is sent back to the SQL server for storage (auditable trail of simulation runs performed) and distribution to the interested clients.

6 User Levels

There are three user levels defined for the system:

- Administrator
- Developer
- Client

The Administrator manages general system functions, such as the type of input information that can be edited, graphics to use for the various ports, user permissions per port and most importantly, approving or rejecting simulation run request from the clients.

The Developer has access to all aspects for the system, but would mainly be required to make ad hoc changes as requested by an engineering request, maintain the system and ensure that the latest simulation models are on the G2 server. The developer will also have control over the default scenario for each port. Typically this scenario is a benchmarked simulation run with the latest information from that port. Whenever a client creates a new scenario, the default scenario's input values will be used as

a starting point. Probably the single most important function of the developer will be to benchmark the various models on a regular basis as discussed later in the paper.

The various clients has access to all the data in the various ports, but can only configure simulation run request for the port(s) for which they are authorized.

7 Configuring a Simulation Run Request

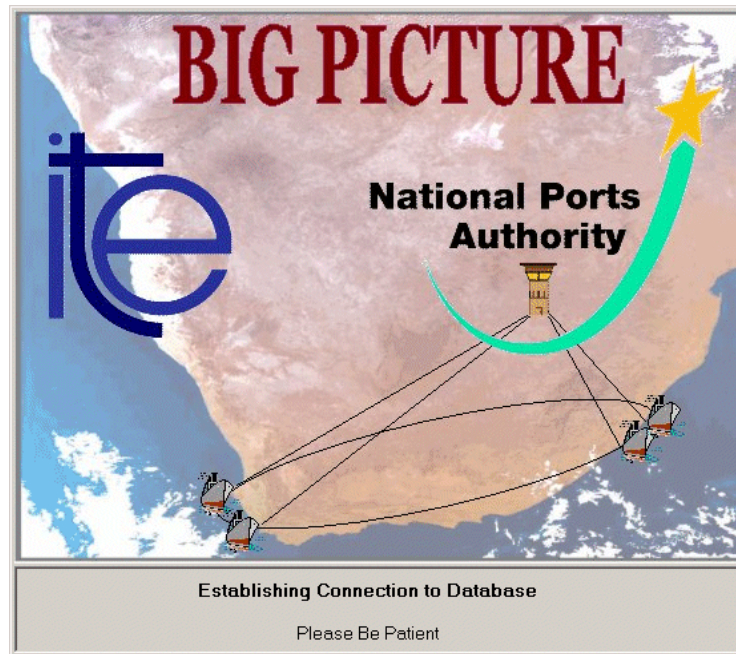


Fig.2: NPA Big Picture Splash Screen

Table 1: Simulation Model included in Big Picture System

Port	Model	Schematics
Saldanha	Multi Purpose Terminal	1.Port area with storage areas, berths, road network 2.Immediate hinterland with road network to various towns and plants 3.Basic rail infrastructure that services the port
Cape Town	Multi Purpose Terminal	1.Berth and storage areas
	Container Terminal	1.Berths, wharf cranes, individual stack blocks, cartage towers and rail-head cranes.
Durban	General	1.All berths, channels, storage areas and rail sidings behind the berths
	Container Terminal	1.Berths, wharf cranes, individual stack blocks, cartage towers and rail head cranes
Richards Bay	Port	1.Multi Purpose Terminal with berths, storage areas 2.Dry Bulk Terminal with conveyors, storage areas, tipplers, vessel loaders/unloaders
	Rail yard	1.Bayview rail yard (each individual rail line simulated) 2.Detailed Nsesi rail yard 3.Detailed Bisolo rail yard 4.Detailed Mhlatuze rail yard 5.Visgraat interface

When the system initiates, the start up screen appears, Fig.2. A ship icon displays each port that is configured in the system. These are listed in

Table 1. Once the system has initiated successfully, a graphical list of the configured ports in the system is displayed, Fig.3.

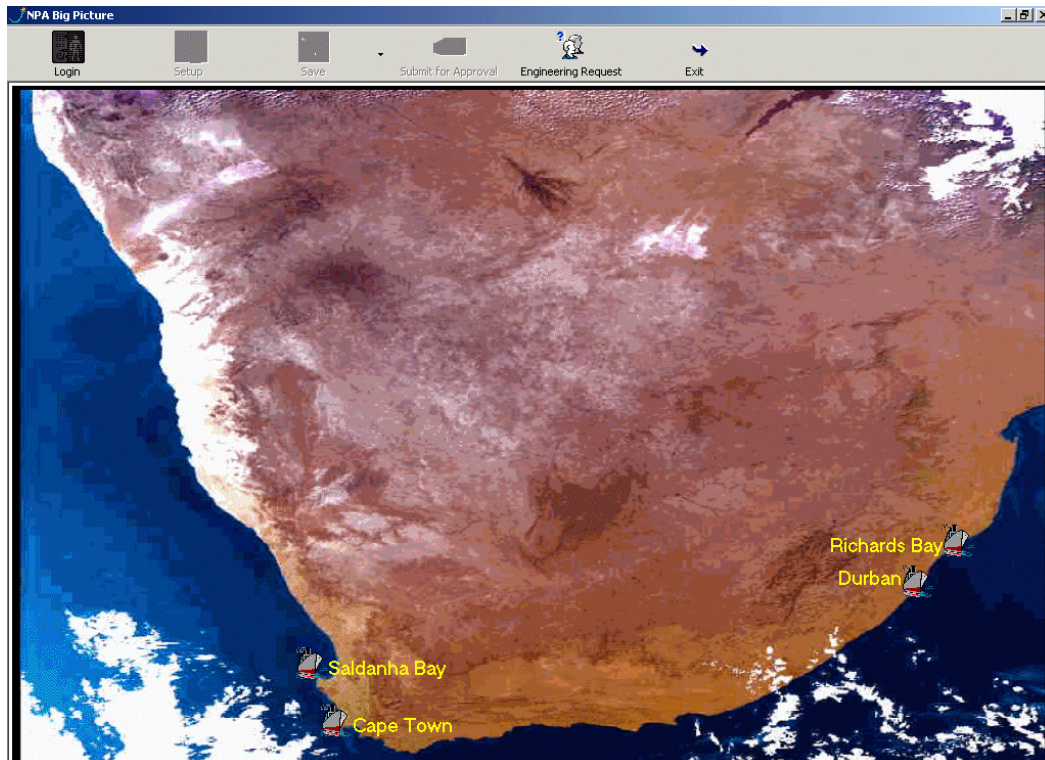


Fig.3: Port Login Display

For each port, a tree view of previous configured and simulated scenarios is displayed, Fig.4. This tree view enables the users of the system to organise the specific scenarios into relevant groupings. All the specific scenarios for a particular investigation will thus be listed under a unique Scenario Group. The status of each scenario is displayed as well.

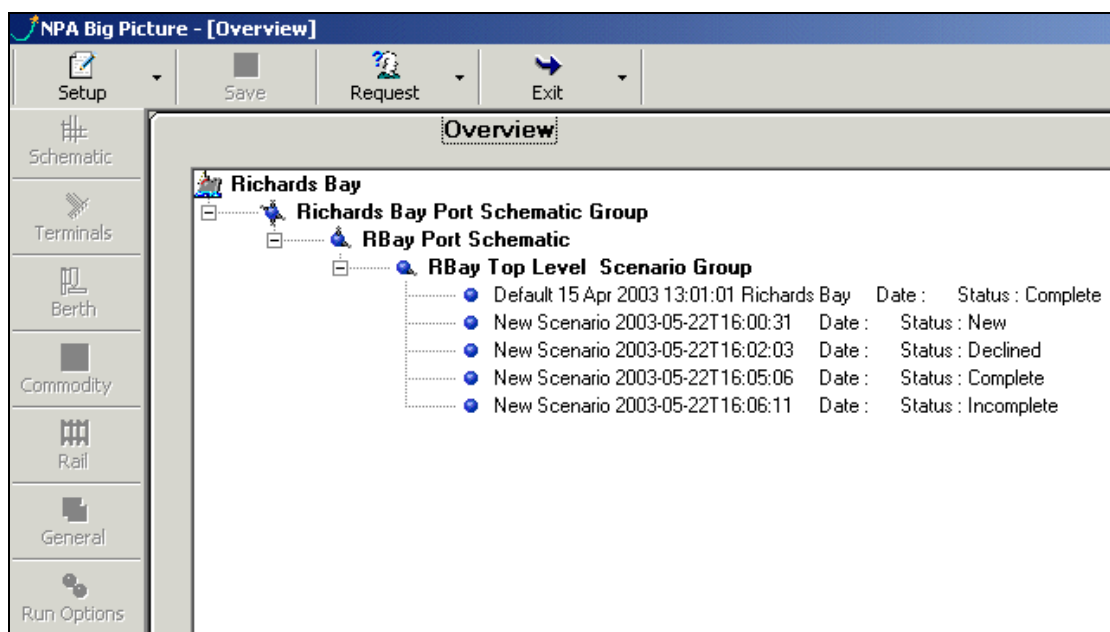


Fig.4: Overview of Scenarios configured for Richards Bay

When the user selects any of the scenarios listed in the tree view, they have additional options to download the results file, load the input information, view/edit the input information or create a new scenario by copying another. Normally the user will create a new scenario by copying the default scenario (most recently benchmarked).

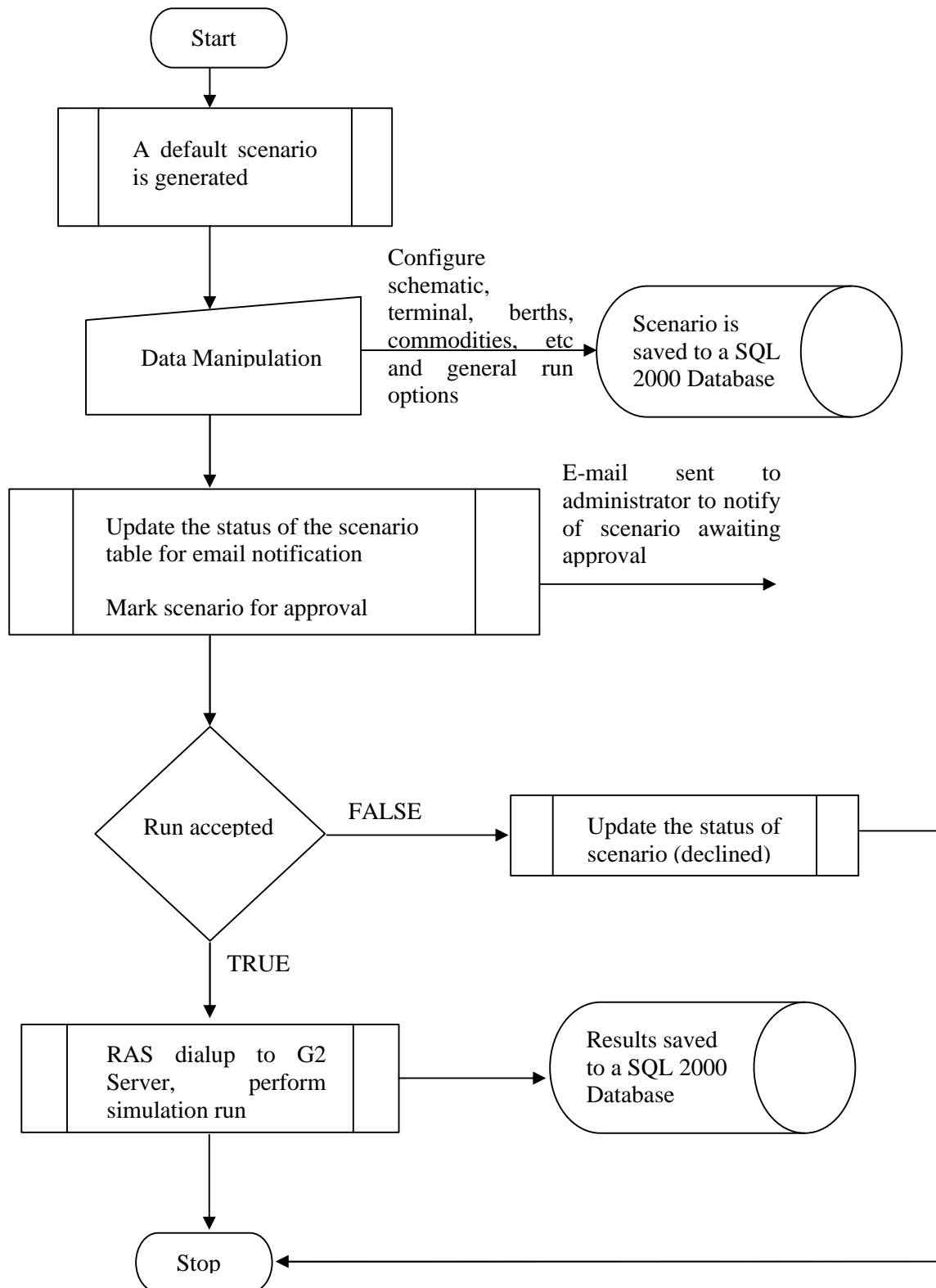


Fig.5: Process Flow for Submitting and Executing a Simulation Run

The user is guided through various input screens where the scenario configuration can be edited. Once all the desired fields are set, the user can submit the run for approval.

An email will be sent to the system administrator to inform him/her of the request that is awaiting approval. The administrator can then log onto the system, review the configured scenarios and choose to accept or reject the simulation run request. If the simulation run is approved, stored procedures in SQL will automatically generate input files for the appropriate G2 models and save them to a specific location on the hard disk. A service on the SQL server performs a daily dial up to the G2 server and pushes the input files across via RAS. At the G2 server, the presence of these files will trigger a service to load the appropriate G2 model, suck in the input file for that specific run, perform the simulation run, write the output file and close down G2. The service on the G2 server will continue with this process until there are no more simulation runs waiting to be performed.

The same service that pushed the input files across to the G2 server, will dial up daily to download the results files (if any) and store them under the appropriate scenario in SQL.

At all the various stages of the process, e-mails will be sent out the client requesting the simulation run. Once the results are in SQL, the client can thus log onto the system and download the result files to his/her local PC for review.

8 Engineering Requests

If there is a specific request that a user has, for instance to add extra infrastructure to a port schematic, (s)he can submit an engineering request that will be relayed to the system developers for attention.

Another example that could lead to an engineering request is when an end user wants to test a different controlling philosophy, for instance giving preferential berthing to a particular shipping line or implementing a slot system for container vessels or using different rules to allocate stack space. All these type of changes to the system can be accommodated, but requires changing the rules and procedures in the relevant simulation model. It thus seeks to extend a simulation study to beyond mere configuration of input parameters. In such a case, the user can lodge an engineering request.

This avenue will also keep the all-important channel of feedback from the end user open, so that the system can be customized to suit.

9 System Roll Out and Implementation

The technical and financial hurdles for this project have both been overcome, but the ultimate success or failure of the system will depend on the ability of the intended users of the system to change the manner in which they've arrived at planning decisions. Unless the users are comfortable with the system, the real danger exists that this might become yet another white elephants of the IT world.

There are two obvious areas of concern to ensure that users are comfortable/competent in using the Big Picture system. Firstly, the users must understand how to navigate through all the options and menus to get the system to perform the functions as intended. For all the ports, the basic system functions, look and feel will be similar with only some detailed information screens that will look different.

The second and arguably more substantial challenge is to train the users in the use of simulation as a planning tool. The challenge here is to get people to change their behavior and specifically their behavior related to making decisions about the future. The benefits of simulation as a relatively inexpensive means to evaluate costly alternatives, based on much more realistic system dynamics than for instance spreadsheet evaluations can be totally negated if the user does not possess a well founded understanding of the underlying assumptions upon which a particular simulation model was built.

Each model was developed for port specific studies and consequently encompassed different levels of detail than the next model. If the simulation results show a specific sensitivity for an area that is simulated very crudely, the danger is that too much weight can be placed on the results and interpretation of those results.

Simulation basically is a tool that paints a picture of an unfamiliar scenario and just about anybody with some basic system training will be able to produce results (a picture). However, the benefit of the successful application of simulation lays in the ability to interpret/read this picture and translates the simulated results to relevant solutions to the real world problems.

Although simulation has made its mark in the planning of South African ports, the next step to put the technology on the desk of every port planner throughout South Africa. The Big Picture system provides the technology that certainly forms a firm base to build on. One of the spin offs of the process of building the Big Picture system is that people throughout the organisation became aware of the technology that is available. Also, there is one central place where operational data is kept. Purely from a reference point of view, the system promises benefit in providing a forum to distribute this information.

10 System maintenance – integrity of information

If the system is to be implemented successfully, it would be necessary to constantly update the information in the system for the various ports and make more ports part of this system. Since each terminal in the ports are basically run as separate business units, the information systems differ from terminal to terminal. There is thus not one standard format for capturing operational data throughout the entire enterprise. At least with the Big Picture system, there is one central database of information that contains the latest financial year's operational data.

It is envisaged that the data in the Big System will be updated annually, hence each terminal will repeat their specific data extraction exercise and this presents strong motivation to establish automated queries, reports and document actions to extract the data from the various information systems. At a later stage, it might even be beneficial to NPA to design a standard information system that can be deployed across all the terminals in all the ports in South Africa, similar to the development of VTSLog – an intelligent database front end that connects radar and VTS technology for capturing maritime information.

11 Extensions to training and development

The Planning and Development department of NPA is considering a formal Port Planning program for the SADC region. The program is intended to cater for the following needs within the NPA and related industries:

- Design & implementation of a paradigm to support Port Planning
- Alignment of skills and a competency audit thereof
- Development and implementation of a resource base and appropriate succession plan
- Ongoing development of a technical skills base
- Professional development and advancement by upgrading existing skills

Currently the skills required by the NPA are gained through various separate courses offered at tertiary institutes such as Universities and Technikons and professional development courses offered by private service providers such as the Port Academy and other technical and business courses.

The current thinking is to present an annual 3 month program using both international and domestic service providers to cover the areas of, amongst others:

- Port Planning & Development
- Port Pricing/Economics/Tariffs/Cargo Projections
- Port Operations/Productivity/Capacity
- Container Terminal Operations/Management
- Port Operations/Planning and Cargo Handling Technologies
- Cargo Transport. Modal Economics
- Institutional Reform/Privatisation
- Critical spill-off developments (logistics parks, etc.)
- Maritimes Economics
- Maritime Law
- Environmental Planning

The aim of the program is to provide a four dimensional process that:

- (a) Provides access for people into the Port (Planning) environment
- (b) Employs and develops the people within the industry
- (c) Advances and recognises achievement within the industry
- (d) Transfers knowledge and experience to others

ITE-C is in the process of tendering to provide a portion of this course, specifically aimed at the Planning & Development portion of this program. For this, ITE-C proposes setting up a localised Big Picture type system in the training room. Students will have a PC in front of them that is networked to a central server containing G2. Over a four-day period, the students will be taken through a complete simulation study, using their PC's to perform the various tasks of the study. ITE-C will set up a simple example port in G2 for this exercise. The simulation study will follow similar objectives as per a majority of the simulation studies undertaken by NPA and ITE-C in the past. For example, the objective of the four days may be to "Benchmark the Model of the Example Port, ascertain the relative improvement of various expansion options, and determine the maximum throughput of the current port."

The following components of the study will be taught:

1. Information gathering – what information is required and how and where to get it. This is not confined to a technical skill but will include people and communication skills
2. Data analysis – how to extract (calculate) information from the data to better understand the performance of the environment (e.g. handling rates, dwell times, delays etc.)
3. Data manipulation – how to prepare information for the simulation model
4. Benchmarking data – how to determine information required to validate the model
5. Output information – selection of appropriate Key Performance Indicators and other indicators for assessing the performance of the port (to be used in benchmarking and evaluating scenarios)
6. Developing a simulation model. The presenter will demonstrate aspects of this for the class – e.g. objects, attributes, parameters, cloning etc. Students will not actually build the model – one will already be built for them to use.
7. Benchmarking the model. Students will be taught how to make sure the model is valid and to determine over what time period the simulation should be executed.
8. Selecting scenarios. Students will be taught how to select the appropriate scenarios and variables to simulate in order to answer the objective(s) of the study.
9. Simulate the scenarios using the Big Picture system (by sending them to the G2 server to perform, after which, the results are returned to the student)
10. Interpreting simulation outputs and analysing results. This will include techniques on compiling the results into a format that supports the study and how to present the outcomes to decision-makers and stakeholders. This will cover the areas of compiling results, preparing reports, and presenting the findings to management. This scope could be extended to include presentation skills.

References

FURSTENBERG, L. (2000), *expert system techniques applied to virtual prototyping and design*, 1st Int. EuroConf. Computer Applications and IT in the Maritime Ind., COMPIT, Potsdam, pp.166-177

KELTON, W.D.; SADOWSKI, R.P.; SADOWSKI, D.A.. (2002), *Simulation with Arena*, McGraw-Hill, ISBN 0-07-112239-7

SIMPSON, D.; FURSTENBERG, L.; KALATHAS, H.; BERTRAM, V. (2003), *Virtual prototyping for developing South African ports*, 2nd Int. EuroConf. Computer Applications and IT in the Maritime Industries, COMPIT, Hamburg, pp.235-246

Knowledge-based Concurrent Engineering in One-off Ship Design

Jenny Coenen, IHC Holland Dredgers NV, Kinderdijk/The Netherlands, j.coenen@ihcdredgers.com

Martin van Hees, Qnowledge, Wageningen/The Netherlands, m.v.hees@qknowledge.nl

Ubaldo Nienhuis, Delft University of Technology, Delft/The Netherlands, u.nienhuis@wbmt.tudelft.nl

Abstract

The paper presents the philosophy behind a decision-support tool for reproducing, evaluating and predicting the 'decision and criteria log file', costs and delivery time of a design process. A demonstration version of the model was built for sub-processes at the IHC Holland NV Dredgers shipyard. The method of incorporating design-knowledge itself in predicting the design process is new and recognized as highly supportive for concurrent engineering by a substantial part of the Dutch shipyards and marine subcontractors. One innovation is using self-mutating calculations to find possible process scenarios, instead of using simulation.

1 Introduction

1.1. Concurrent engineering in shipbuilding

Fig.1 shows a typical process model the building of a ship at a Dutch shipyard, illustrating the main information exchange between owner, yard, and subcontractors. The important process steps are arranged in time order (top to bottom). The project starts with a preliminary study of the owner (with possible participation of a yard and subcontractors), resulting in a request for proposal (RfP) for one or more yards. Then the design work follows, involving preparation of contract negotiations, the contract and specifications. In case of a contract, 'systems design' is done by field specialists, with input from subcontractors, contracting of subcontractors and customer approval of systems design. After approval, working drawings are made in cooperation with subcontractors to produce an approved engineering package including all necessary production information. Further steps (production, etc.) are beyond the scope of our research. Fig.1 gives only a global overview. To reflect the concurrent and consecutive nature of the process, the figure should show a separate process for each main ship system and engineering discipline. Then these models should be linked as they all occur more or less in parallel with mutual information exchange.

Preferably, we would build ships in a structured, phased way: Starting a new sub-process when the preceding process has been finished. However, owners desire short throughput times (at low cost), and the capability of Dutch shipbuilders to realize these short times is one of their great strengths. But this leads to the risk that the costs of correcting (even small) deviations of the planning can mount high... The whole shipbuilding process, but especially the design process, suffers from severe knock-on effects that result in disproportionate costs for setting off mistakes that have been made earlier in the process, mostly due to severe time pressure.¹ The main challenge with respect to successful shipbuilding lies in recognizing these knock-on effects and using this knowledge to set priorities properly: what activities are under 'real time pressure' and critical regarding their impact on the total throughput time and the total costs?

Understanding this domino effect is not easy. Even an attentive observer will only get a troubled view on process structures. Imagine a shipyard manager explaining to visitors 'the shipbuilding process'. He will probably take them to the slipway and workshops. It is not enough for understanding the complete production process, but it is sufficient to get an impression of what these people do, what work is at hand, the demarcation of processes. Further observation would give information on the

¹ The reason for this large influence of design on the total costs can be attributed to several factors: first the substantial amount of hours that are spent in design stage (to get an idea of the order of magnitude, think of hundreds of thousands), and their spin-off in production (estimated at about 5 times the engineering hours), as the quality of design affects strongly productivity, and their spin-off in purchasing, as the quality of specifications is of great influence there (purchase covers typically 80% of the total costs)

sequence of actions and necessary time. A smart observer could recognize bottlenecks as well. For a further glance at the process much documentation is available like safety procedures, simulation programs for machines, ISO certifications, QA/QC-manuals etc, or can be made using established off-the-shelf methods. For the design and engineering process, this is more complicated, as observing the production process potentially gives more insight into process information than looking at people staring at 21 inch monitors or making telephone calls.

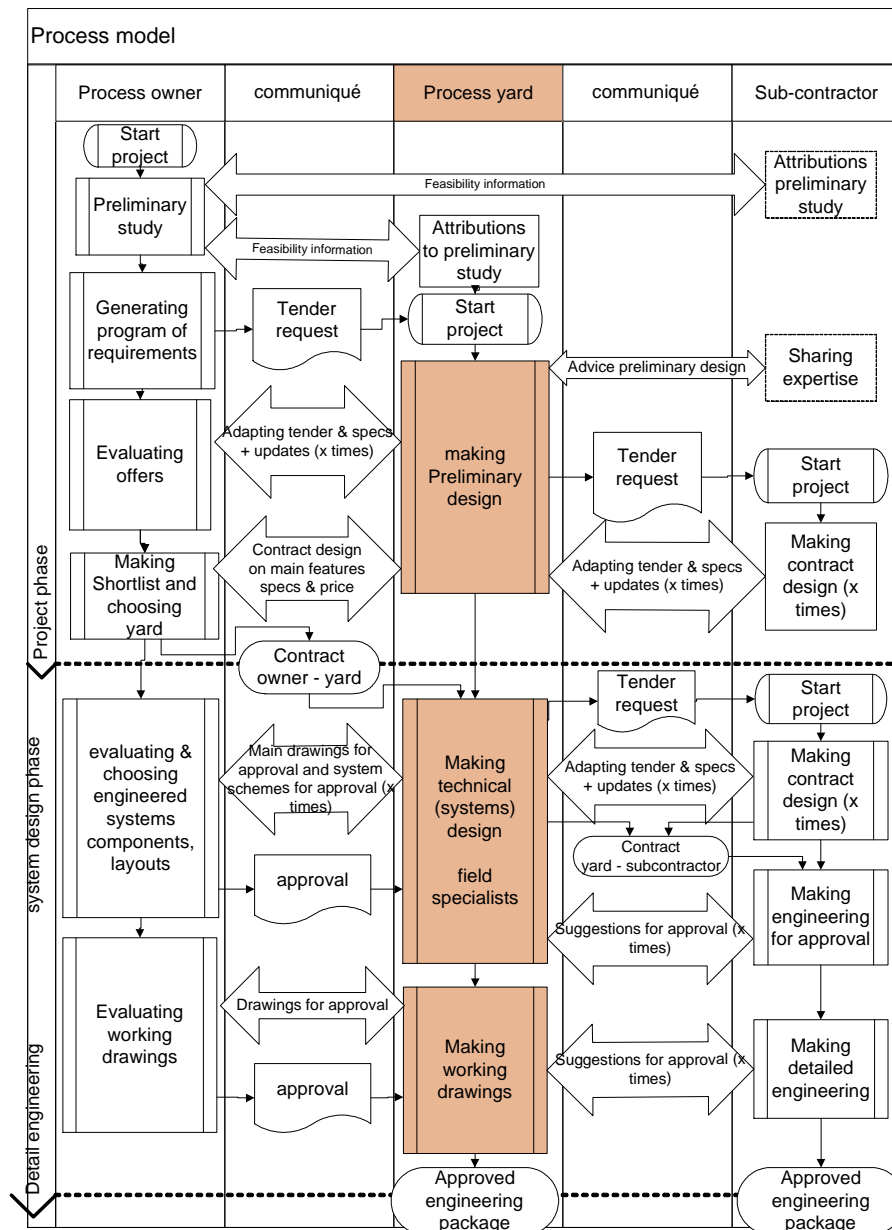


Fig.1: Process model of Dutch shipyards

What are designers doing? In an abstract sense, they perform a series of activities that take a design problem from the initial specification to several sub-artefacts that after integration still meet all the specific and overall requirements. As the amount of existing subsystems on a ship is high (transportation function, storage, housing and all the supporting functions; for IHC Holland also dredging), this process is fairly complex. Many people work at the same system or space simultaneously (concurrent engineering!) and constantly update each other's system environment and product information. Imagine the equivalent situation in production: one engine room section of 15 meters length, 10 production employees of the yard, six fitters of subcontractors and a purchaser. Each person is constantly fitting other components, keeps changing the connections and foundations,

or puts components aside to make more space for others, or changes dimensions and capacities, with a few changes in customer demands to incorporate too. Nobody would be surprised to hear that this section was built according to Murphy's Law at 300% of the costs and expected lead time. But in fact, the chaos in the virtual building process that is called engineering comes close to this. Miracles apparently exist (as all these ships are launched nevertheless); all this is accepted in order to have a short throughput time. (And after the launch engineers and designers start good-humored with the next example of craftsmanship and cool-headedness). It sounds paradoxical to build in so many dependencies in a process because of the potentially disastrous effect on lead times. However, doing everything in parallel is the only way to realize the short throughput time in engineering required to maintain the production capacity at the requested high level.

1.2. An approach to researching concurrent engineering and design processes.

A logical first step in throughput time related research is to make an inventory of the existing situation and model it, in order to be able to replicate the behavior of the process. Then hypotheses on influencing the behavior of the engineering process can be tested before setting of with tests in the real world (which is advisable as most yard managers will not sympathize with the idea of using the real yard as a laboratory). As illustrated in the previous section this inventory making is more easily said than done: really understanding the actual situation is a sizeable job in itself. It asks for meticulously following of all the strings that bind the activities of these engineers that all work together on the same few meters of ship. Modelling of engineering process behavior has been researched earlier on small scale, *Cho and Eppinger (2001)*, but as shipbuilding has added at least two extra dimensions to the concept of concurrent engineering², some new challenges have to be faced in this field. If this resulted in a functioning concept for modeling and reproducing process behavior, it could then be used for process monitoring and managing too. If modeling is possible research on improving throughput times and costs can be set up.

1.3. Scope of the research and demarcation

1.3.1 Identification and registration of process ingredients

The first objective of our research is to identify the main ingredients of the ship design, to disentangle the 'spaghetti' in the design processes. The activities, relations between activities, interaction between the design activities at the yard, and the interaction between design offices at the yard and subcontractors are inventoried and visualized. One application of this knowledge, when gained, should be to evaluate the ship design processes and suggest improvements. As the ship engineering process is not set up structurally, but grown quite organically, this first objective will require substantial effort.

The results of process evaluation with respect to cost, quality and delivery time should be measured from start to finish (covering the whole ship creating process ideally). Without anticipating on the practical modeling problems arising from attempts to cover the complete process, it seems reasonable to choose a pragmatic demarcation of the parts of the process that are modeled first. The yard management has expressed an interest in the moment of delivery of the 'approval information' resulting from systems engineering, as it is believed to effect process control in detail engineering substantially (and this in turn effects the possibilities for successful pre-outfitting etc. The moment of

² We suggest five dimensions of which the first three are alike for all concurrent engineering; 1-consecutive activities; 2-sub-goals per activity that may be competitive and have to be harmonized for the overall goal (Dutchmen are likely to take this concurrent aspect too literally, as the Dutch word for competitive is 'concurrent'. This defending of own sub-goals by fire and sword further complicates matters though), 3-shorter throughput time. The fourth and fifth result from typical shipbuilding methods: 4-many involved subcontractors with partial product responsibility and the resulting information cross flows; 5-pre-outfitting which asks for both system and space engineering in an early stage. Another may be added for the future: the further layer of having a sizeable time-overlap between engineering and production.

finishing systems engineering is regarded as a first indication for influence on general throughput time in detail engineering and production. This identification objective was recognized as highly supportive by most Dutch shipyards and marine installation companies and subcontractors³.

1.3.2. A note on visualization of process models

Visualization of the design process is not an objective in itself but serves different purposes:

- The gathering and recording of information from yard sources and experts profits from clear visualization methods.
- It makes (cross) validation of the gathered information easier as long as the pictures are interpreted in the same way by all parties.
- Illustrating the modeling of relevant information in order to replicate the behavior of the (modeled) process.
- It serves as an input-interface for a prediction tool.
- Illustrating time aspects as lead time and spent hours are important measures for process performances.

Not all purposes are served equally by different available visualization techniques. Illustrations for discussion and information gathering and validation should be simple, clear, unambiguous and still incorporating all relevant aspects and subjects of discussion. The problem is known from other fields of software engineering, and several model techniques have been developed for visualizing process models.

At this moment, process information is translated manually into input for the prediction tool. The information is stored by means of 'knowledge rules' that describe the relations between objects. These objects can be processes and documents. This can all be covered by the used symbols in the flowcharts as described, so this visualization method also serves interfacing purposes.

This leaves the time aspect which is not covered in a conventional flowchart. A rough improvised method to illustrate time aspects is leading the flow along a timescale to get an idea of starting times and end times. This is of course not very precise as iterations in the flowchart are not according to the timescale and the size of the blocks may also suggest time aspects that are not present. A PERT or Gantt chart can be used to overcome these problems.

In Fig.2 the task 'check available space' is marked. According to the flowchart there are two possibilities to start this task. Scenario a: the task starts when the task 'make component list' has finished. In an alpha-numerical way this could be represented by:

Check available space=make component list

This is only a 'one-way' function, but not an equation, as it cannot be read from right to left (thus not like $F=m*a$, which could also be written as $m=F/a$ or $a=F/m$). (The '=' sign is in fact an ill-chosen notation, which may be replaced in a later version) In scenario b, the task 'check available space' can also be preceded by 'check changes in other parts' or:

Check available space=check changes in other parts

Further on, the necessary input elements can be enclosed in the same notation system:

³ Being united in the research project CE3P (Concurrent Engineering and the effects in Production, Pricing and Planning).

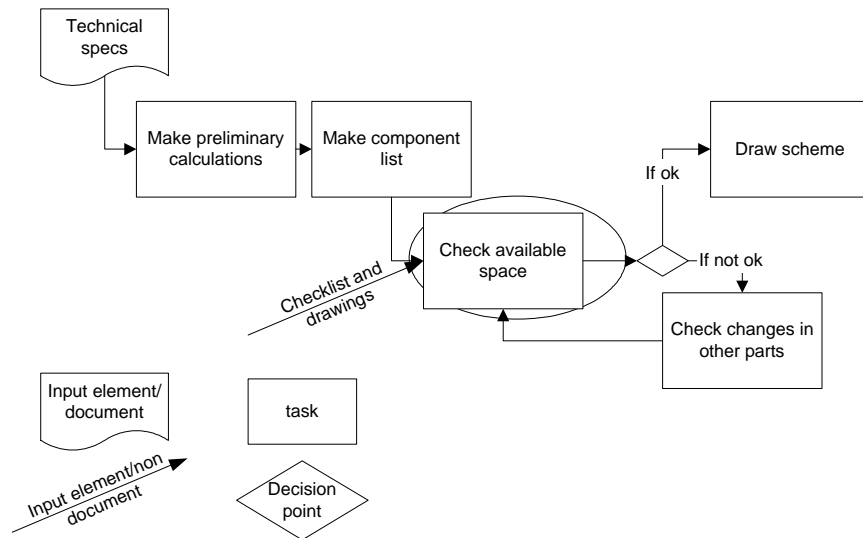


Fig.2: example of the use of flowcharts for design process modeling

Check available space=make component list+*checklist&drawings*

The expression used above for the starting of a task (check available space) implicitly suggests that making the checklist and drawings is not a part of the process. This can mean two things: the list is made outside the process, or the list has been defined as the result of the task ‘making the checklist and drawings’ which has been defined elsewhere in the knowledge base. The algorithm that tries to solve the expression will reason that if it cannot find this definition it is an external element. If it can find a definition for the element ‘checklist&drawings’, this will have the following form:

checklist&drawings (hardware)⁴ = make checklist and make drawings (task).

Then the expression for ‘check available space’ will be equivalent with:

check available space=make component list+make checklist and drawings

The knowledge rules can now be stored in a very compact way in a database. The further editing can be done directly in the alpha-numerical representation which is easy to use for a reasonably trained reader (comparable to algebra), *Coenen and Nienhuis (2003)*.

1.3.3. Developing a research tool for measuring the effects of intervening in the process

The second objective is a research tool as an intermediate station to the ultimate objective: process improvements in concurrent engineering. The tool is henceforth called Process Evaluator and should be able to illustrate the characteristic mechanisms in ‘the process of designing a ship’⁵, reproducing process behavior, allowing to carry out changes in the process or process characteristics and comparing the results. The tool should measure the throughput times and costs as mentioned in section 1.3.1. The tool is initially meant for research purposes only so the demands regarding user-friendliness are less strict than for an implemented support tool. Such a support tool would be an interesting, though futuristic spin-off of the research tool: a concurrent engineering wizard that processes input (in the form of a process model with relevant product information and historical data) into network critical path analyses, business process alternatives, tracking possibilities, automation of engineering, in fact a complete HAL9000⁶ for modern shipbuilding. This tool could then be used for progress measurements during projects (process management), information, data, and knowledge management, improved scheduling, and evaluation and prediction of process changes.

⁴ Hardware: objects with a ‘physical meaning’ like people, drawings and documents

⁵ the term “process” from now on implies the whole of activities/tasks/sub processes involved in designing one ‘Contract Order’ (a single ship)

⁶ Referring to the classical movie and novel by respectively Kubrick and Clarke, ‘2001, a space odyssey’

1.3.4. A preview of the conclusions on design automation

The third objective is to show that the concept of using product knowledge to help predicting the design process is useful indeed and that the Process Evaluator is a suitable shell for implementing design automation tools. An attempt will be made to couple a “product configurator” (rule based reasoning) to the process evaluator. The results of the research ‘Knowledge and Data Reuse in Ship’s System Design and Engineering’, *Nienhuis and Nieuwenhuis (2004)*, are watched with great interest though, to possibly implement other methods of knowledge reuse as well.

2. Process modeling observations

It is difficult to distinguish between the ‘contents’ of the engineering process and the process steps themselves. The reason why the ordering of an airco unit necessarily follows the calculating of the necessary capacity is as obvious as for calculating the necessary capacity after calculating the contents of the room to be air-conditioned and the necessary amount of air changes. The ‘numerical rule’ for calculating the capacity [$necessary_cap = n(airchange) * V(room)$] would also be useful to describe the process rule which says that calculating the capacity must be carried out after the tasks *calculate amount of air changes and volume of room* have finished. Examples of this kind cover a substantial part of the description of the engineering process. Thus merging two types of engineering rules (the ones concerning the ‘contents’ and the ones concerning the ‘scheduling’) should be possible.

The chosen level of detail should be such that the behavior of the model represents the behavior of the real-world processes, such that it distinguishes between the tasks that affect the behavior of the process substantially and those that do not. This is not a very useful rule for choosing detail level as the model should be built (and the level chosen) before results are available to check the behavior. So trial-and-error is needed. The model can be further zoomed in (or out) as necessary. For now the level is chosen in which all information transfers to different resources can be shown.

Often, several alternative relations exist between tasks, e.g. as a result of iteration in the process. Iterations can be related to design/product or to organization. Alternatives often depend on decision points in the process, like in Fig.2 where the task ‘check available space’ should result in concluding whether there is enough space, or not. If the answer is ‘false’, another route is taken through the process flowchart than if the answer is ‘true’. A question is how these loops are dealt with, as the example in Fig.2 does not give any information on the maximum amount of loops or the shifting of probabilities of outcomes between following loops. This has been settled by defining each task that is an instance of an earlier task, separately.

3. A Process Evaluation and Prediction tool for concurrent engineering

3.1 The basic choices regarding functional specifications of the tool

The basic inputs for the Process Evaluator are the tasks that form the sub-processes of the engineering process and the relations between these tasks that set the order of the tasks. These tasks have attributes as for instance the necessary capacity for a task and can be linked to product information that is in some way connected to this task. The output should be a set of alternative process networks, ranked according to their desirability of occurrence and their expected probability of occurrence. The output should give feedback on the possibilities to influence this desirability/occurrence balance. Desirability should be expressed in costs and throughput time initially at least for engineering but ultimately including as many spin-off effects as possible...

3.1.1. Finding analogies in production related research and discrete event simulation

Earlier knowledge-based research regarding the performance improvement of (production) processes covers among other things the following examples. *Stidsen (1996)* tried solving job shop problems using Genetic Algorithms. Advanced techniques from operations research can be used to combine all possible events regarding the customer, his order and the manufacturing environment, *Wullink (2002)*. Hybrid scheduling systems exist that integrate different scheduling techniques, knowledge for scenar-

ios, and knowledge about the design of scheduling systems, *Saurer* (????). Examples of dynamic planning, in which a plan must be continually updated in light of changing operating context can be found in spacecraft operation.

A discrete event simulator models the system of interest by entities that flow through different process steps. The processes cause a delay in the flow of the entity and they require the availability of time and resources). After the process has finished, resulting in a certain transformation of the entity, the entity is released to the flow. E.g. a crate with empty bottles arrives at a certain time t at the filling station. The entity is first split into 24 separate bottle-entities (this takes some time). Then the bottles are cleaned and rinsed (process); then filled (process), closed (process) and put back into the crates (joining and process). The crate leaves the station by truck. The complexity of these processes is reduced by the integer-nature of the entities. The crates and bottles do not normally come in parts.

Now imagine a virtual entity like a document called 'customer ship wish list' that has to be transformed to a package of entities like documents that are called 'detailed ship production info part a'. There would be many parallel flows in a simulation model for this as the entity is split many times into very small, arbitrarily sized, virtual entities and these are often joined again for producing document entities. The arrival of this wish list entity marks the start of the systems design engineering at the drawing office of the yard. Using a commercial-off-the-shelf discrete event simulator could be considered to be used as a base for the Process Evaluator. They all have built-in features for including randomness with respect to process duration and the chance on a route for the entity to take. This allows to consider aspects that cause the total throughput time of the flow to be different each time, if these built-in random factors are capable to represent the behavior of the process properly. Nevertheless, because of the limited possibilities for linking the process information to product information and the limited possibilities of product and process knowledge reuse, another base was chosen: Quaestor.

3.2. Knowledge-based Computational Model Assembling (KCMA)

3.2.1. An introduction to Quaestor

Quaestor is a knowledge based computation model assembling tool on the basis of user provided data and a knowledge base containing computational model fragments such as formulae, program interfaces and design data (*Knowledge-based*), executable computational schemes (*Computational Model*) can be constructed (*Assembling*) for an important class of design and analysis applications. As the output of one formula can be the input for another one, the formulas form a network of rules. During problem solving, the designer provides the objective, and Quaestor suggests formulas or methods to be used to solve the problem. Quaestor will prompt the user for values of parameters applied in the proposed relation. Values that are known can be calculated or filled in. By either accepting or rejecting these suggestions, the designer selects the rules and thus fully controls the way the problem is solved.

3.2.2. Processing and storing 'business process' information

An important concept within this and related research is that of function and function-performer. A function states the objective that has to be 'delivered'; and the performer is the mechanism that supplies it. In knowledge technology of Quaestor, the performer for the information delivery is an automatically assembled set of available knowledge rules, tools and data. A function performer is a model that consists of a certain amount of sub-models, that are gathered by the knowledge system that balances supply and demand.

The objects in the knowledge base can be information, or data, or a design, but as it is possible to attach any attribute to an object, this may also be a task, or an activity. And then labor (time and resources) is a function-performer (e.g. for the function 'pumping' an object pump with attribute capacity can be the function performer) This implies that uniting product and process information is indeed possible and that both planning and engineering are comparable balancing problems. Any function introduced in the model should be fulfilled, and other functions that are necessary to fulfill

this function, may require other functions, and this goes on until the start conditions have been fulfilled.

3.2.3. Avoiding discrete-event-simulation by mutating solution models

A feature of the knowledge system that is introduced as a result of this research is the possibility to generate a ‘initial solution’ first, which may be followed by generating all alternative networks from the alternative relations within the knowledge base. The ‘first network’ choice is based on the internal criteria for the ‘best solution’, which in most cases will coincide with the simplest network. But objects and rules in the knowledge base may be defined by different relations: recall the decision diamonds in event simulation with a chance on exit b and on exit a. The different exits of the diamond are covered in the knowledge base by a relation between input and exit a, and a second relation for input and exit b, Fig.4.

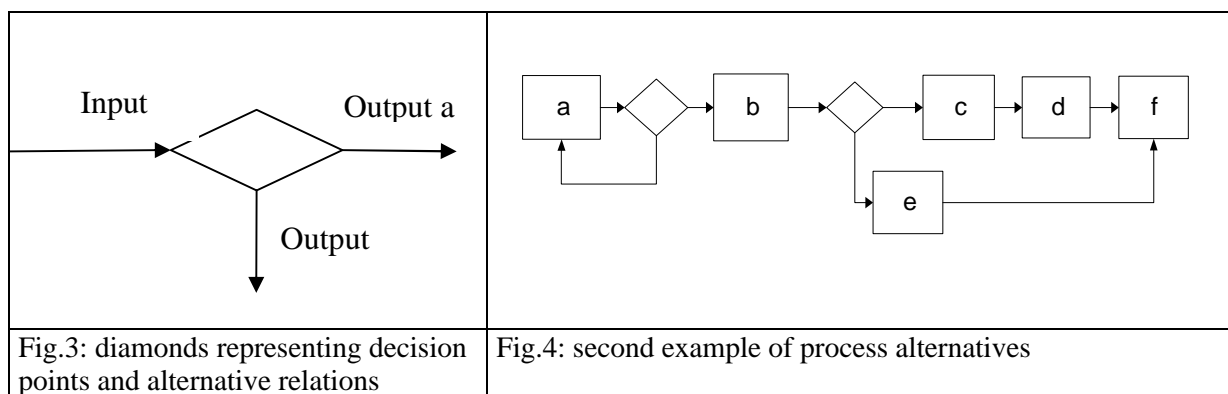


Fig.3 shows a decision point in the process, of which is known that the chance on output **a** is 30% and on **b** 70%. An event simulator would use this 30%-70% division to draw the exit from a virtual set of 30 white balls (**a**) and 70 red (**b**). Using a sufficient amount of runs the expected outcome will be that in 30% of the cases **a** has been chosen. (In this example only two-way diamonds are used. Multi-way diamonds also exist but can always be reduced to a combination of two-way diamonds)

Quaestor will not search in its first run for the most probable option, but the ‘best’ option, and will choose output **a** or **b** on an internal criterion. The first network is not essential, only a start point for further processing, after it has been generated, the solution is mutated in order to find all possible alternatives from combining all remaining information in the knowledge base in all different ways. So in this case the mutant function will find a second network that includes option b.

A slightly more complicated example, Fig.4: Quaestor finds e.g. **a-b-e-f** (option 1) as first solution.

And after mutating: **a-a'-b-d-f** (option 2);

a-b-c-d-f (option 3);

as well as **a-a'-b-c-d-f** (option 4)

In this case the maximum number of iterations for **a** has been set to one loop.

If both diamonds have a 50/50 division for the exits; then the distribution of the chances for a certain process network (option 1, 2, 3 or 4) would be 25% for each.

A discrete event simulator would find the distribution after 40 to 100 runs, Quaestor will generate four networks and calculate the distribution afterwards.

For this research the mutant variant is preferred to the simulation. The main reason is that for applying the value of a ‘task occurrence probability’ (equivalent to the chance on an outcome of a diamond in the process), knowledge of the complete form of the process network gives additional information. What happens in the process may influence the chances before and after the event that is under consideration. (This is difficult to explain without relevant examples, though). First generating the form of the network (regardless of the fact that it will ever occur or not) results in a ‘pseudo’ certainty on the activities that are carried out and the necessary tasks (it is certain within the scenario, but if the scenario itself occurs is uncertain). The other reason is that when chances/probabilities of single tasks

are applied afterwards, they can be calculated within the knowledge base during network generation and related to product information in the knowledge base.

3.3. Methodology and building bricks of the tool

The solver can approach a knowledge base with relations between sub-processes and tasks. This can help to generate solutions in the form of activity networks. Fig.5 illustrates the architecture of the evaluation tool. Part A shows the ‘normal’ architecture of Quaestor. Via the user interface the designer and the Modeler combine facts, rules and input to find a possible solution for a balancing problem (for a design or planning). This is the first step to process evaluation. As we are interested in different possible outcomes of this evaluation, all alternatives of this first activity network are generated by means of the mutation algorithm, see Section 3.2.

The probability of occurrence of an alternative network or process scenario is a function of the chances of individual tasks to be part of a solution. As already explained this individual chance can be assigned to all tasks after all mutations have been determined, see Part B of Fig.5.

Fig.4 already showed how the chance on a scenario was calculated from the chances that a, c or e show up in a solution. Not all ‘task occurrence probabilities’ are relevant for the distinction between different scenarios as some differences on task level do not result in relevant differences in total throughput time.. Even for a few task alternatives (modeled by the diamonds), the number of scenarios can easily amount to a few hundred. The task occurrence probabilities depend on factors discussed later.

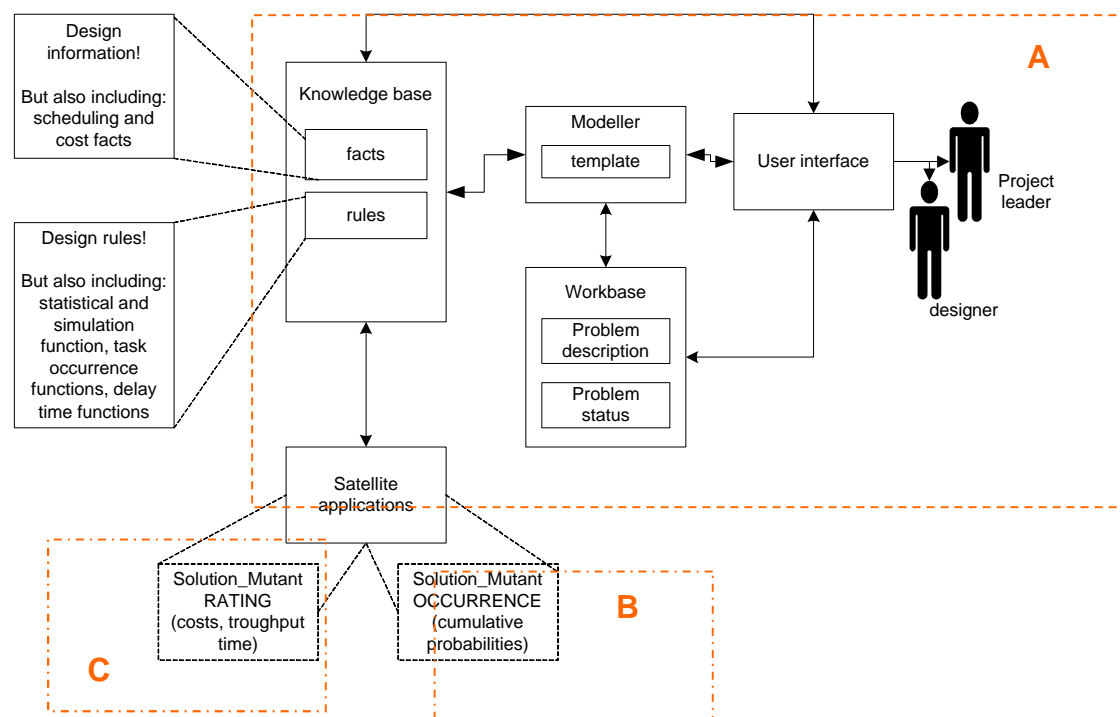


Fig.5: architecture of [Q]-business process MUTANT HERO

The other step after mutant generation is to value all scenarios (C). For throughput time evaluation, this is done by applying all delay time information to each separate task, and by providing information on the available capacity. A planning function in Quaestor balances the requested capacity and order of sub-processes necessary for the evaluated object (e.g. a milestone in the process). The delay time of a separate task depends on several factors like product dimensions or skill of the engineer. There are also random aspects involved. Thus the delay time has a certain statistical behavior itself. In a comparable way labor costs can be calculated.

The values versus the occurrence of a scenario are evaluated by generating histograms that show the cumulative chance on a range of lead times. This information can be helpful to carry out sensitivity analyses for finding critical process schedule bottlenecks in the model.

3.3.1. Complicating matters: Multiple influencing by factors

It is assumed that task occurrence chances and delay time per task may vary for each 'contract', due to order- cq. ship related characteristics and 'other circumstances'. These two concepts together explain the observation that customer driven development appears to be different for each order.

'Product related factors' can be defined as characteristics defined by the contract parties and corresponding technical specifications, and all matters that are directly related to these contract specifications. An example is the *total requested power output* that might be an indicator for the necessary man-hours for drawing the layout of the switchboard room. But *total requested power output* could also be an indicator for a 'task occurrence chance'. It could be assumed e.g. that for ships with a 'high' power request the chance on iteration steps in engine room design is higher than for ships with a 'normal' power request. It implies that this product characteristic should be taken into account in both 'solution rating' and 'solution occurrence'.

Then there is the rather vague concept of 'other circumstances' that does not relate to the contract in its broadest context, but to the 'process environment' (Summer holidays, many projects at hand, global economic situation, voodoo (!!!), (un)lucky stars, experience and know-how of the engineer involved etc). A normal business environment will be conditioned to limit these ranges as far as possible, but this does not completely eliminate their existence.

We assumed that product related factors can be quantified as functions of contract characteristics, while the 'other circumstances' can only be recognized as 'existing' and estimated in relation to the other factors. Together this leaves 4 types of process factors:

- product characteristics influencing task delay times and other internal task measures
- product characteristics influencing task occurrence probabilities
- environmental conditions influencing task delay times and other....
- environmental conditions influencing occurrence chances

3.3.2. Task occurrence chances

The *task occurrence chance* is defined as the probability that a task (or sub-process) is used in a process network configuration (i.e. is part of a (mutant of) a solution found by Quaestor Business Process Modelling algorithm). This is equivalent to the probability of the exit of a diamond leading to the particular task in a flowchart representation of the process network. Recall figure 3: the task occurrence chance of **b** is 70%, namely the 'exit-chance' of exit 2 of the diamond (task **b** is defined as *b after a*, using the loop of exit 2 gives *a₁ after a*, and *b₁ after a₁*. (*a₁* is an instance of *a*, while *b₁* is an instance of *b*). Prominent examples of diamonds and iterations are those that follow 'synchronisation tasks', in which information is compared and geared to one another. For instance: 'evaluating an offer' can have two outcomes: 1) *the offer is ok for all parties; go on with process*, or 2) *offer is unacceptable (incompatibilities in expectations of price), go to tasks preceding Making an offer*" The same is often found for the layout of components, where all concurrent parties have to agree on the location and volume of components in an area.

Fig.6 shows an example flowchart containing tasks that have to be carried out at the yard at the moment of transferring information from the pre-contract design department to the engineering department. The two steps carried out before contract were calculating the capacities of the HVAC installation (1) to estimate the necessary components (2), which are both necessary to estimate a price. Then these data are transferred to the engineering department and in systems design the calculation is reviewed and compared with new design information, resulting from other steps in the engineering department (diamond 2).

Factors that might influence the probability of the outcome of the diamonds can be illustrated with a representation that is equivalent with a 'fault tree' known from risk analysis, Fig.7.

A distinction can be made between two types of factors (product and environment). We already decided that environmental factors (blocks with heavy weight borders in Fig.7) are indeed causing a certain randomness on the outcome of events. Although difficult to estimate their influence it is important to get an idea of their effect on the total outcome of the process as some of these factors can be manipulated by introducing new approaches (like emailing instead of faxing, automating standard

calculations etc.) The product related factors themselves (underlined in Fig.7) such as the *huge amount of (electrical) consumers in a ship* or the amount of times that *the customer normally changes of thought* can be measured objectively, but translating this into a ‘task occurrence chance’ function is still difficult. These factors can often not be manipulated to improve the process, only recognized.

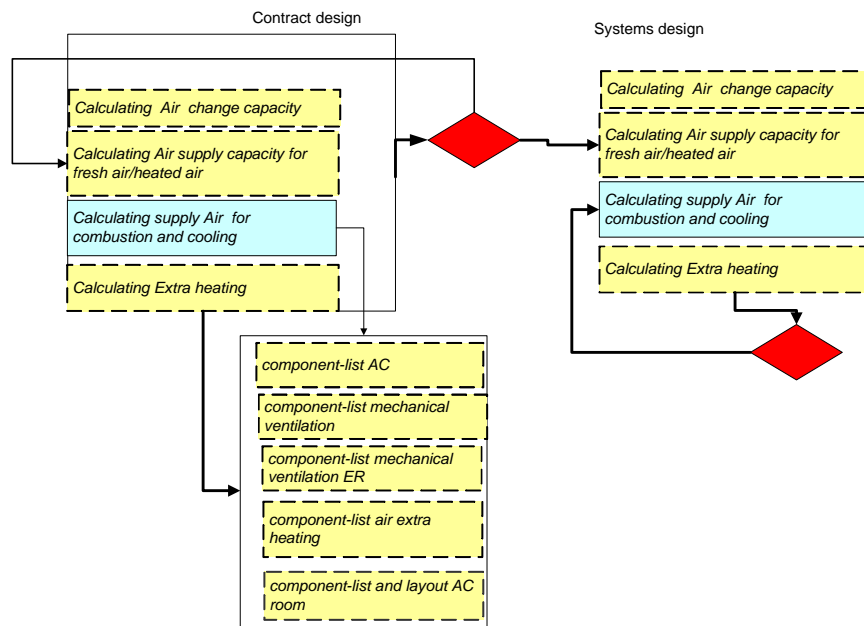


Fig.6: example flowchart for including synchronization tasks

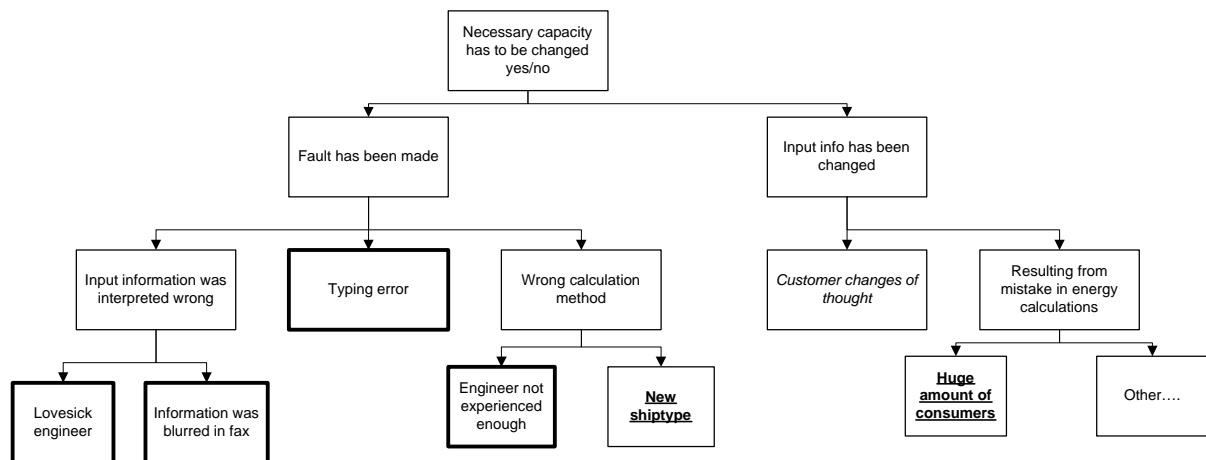


Fig.7: “Fault tree” for inventory of factors that influence task occurrence

The main information sources useful for quantifying the task occurrence probabilities are expert estimation; to some extent historical data like the hour registration for design activities; and reports on design decisions. To quantify this subjective information, techniques developed in social geography can be used, *Timmermans and van der Heijden (1983)*. With historical data and a rough inventory of total task occurrence chances, a first sensitivity analysis should be carried out to determine which diamonds are relevant for the total process behavior. For each relevant diamond the indicators for the probabilities of the outcomes should be described (e.g. by this fault tree method). The range of probabilities relevant for the total behavior should be concluded from the sensitivity analysis. The factors that are mentioned should be weighed. Concentrating on the product related chances, these indicators and their weights can be plotted in different combinations against the observed chance (from historical data). Methods for filtering the environmental chance out of the observed chance are still under development. In social geography earlier attempts tried to explain ‘choice-patterns’ from subjective expert judgment. It is assumed that a ranking list of factors considered by experts as negative or positive, is not always the result of an objective qualification, but is also influenced by

perception, combination rules (for weighing priorities) and decision rules. The developed method for quantifying decision behavior of experts is also of interest in this research.

3.3.3. Randomness in scheduling

Measuring process performance (solution rating in Fig.) in throughput time asks for a scheduling algorithm applied on the networks as generated by [Q]uaestor as solutions of the evaluation problem. This is a critical path analysis for a set of pseudo-certain tasks. The pseudo-certainness results from the fact that uncertainty on the occurrence of tasks were eliminated by dividing the evaluation between rating and calculating probabilities. For each evaluated solution it is assumed that this certain set of tasks is part of the solution. The only necessary information for scheduling is then: order relations between tasks (already defined in the network), and delay time for a task (which results in most cases from the requested capacity and the available capacity in resources). It is good practice though to include a randomness aspect that represents the influence of the earlier mentioned environmental factors that also affect the task delay times. The product related factors should be incorporated in the determination of the requested capacity. Environmental factors or equivalent influences are also known in further developed fields of process evaluation and often taken into account by a normal distribution of requested capacity or delay time, or a triangle distribution. Still the bandwidth of the distribution has to be defined specifically for each case and this can be estimated in the same way probabilities for environmentally affected decision points are estimated.

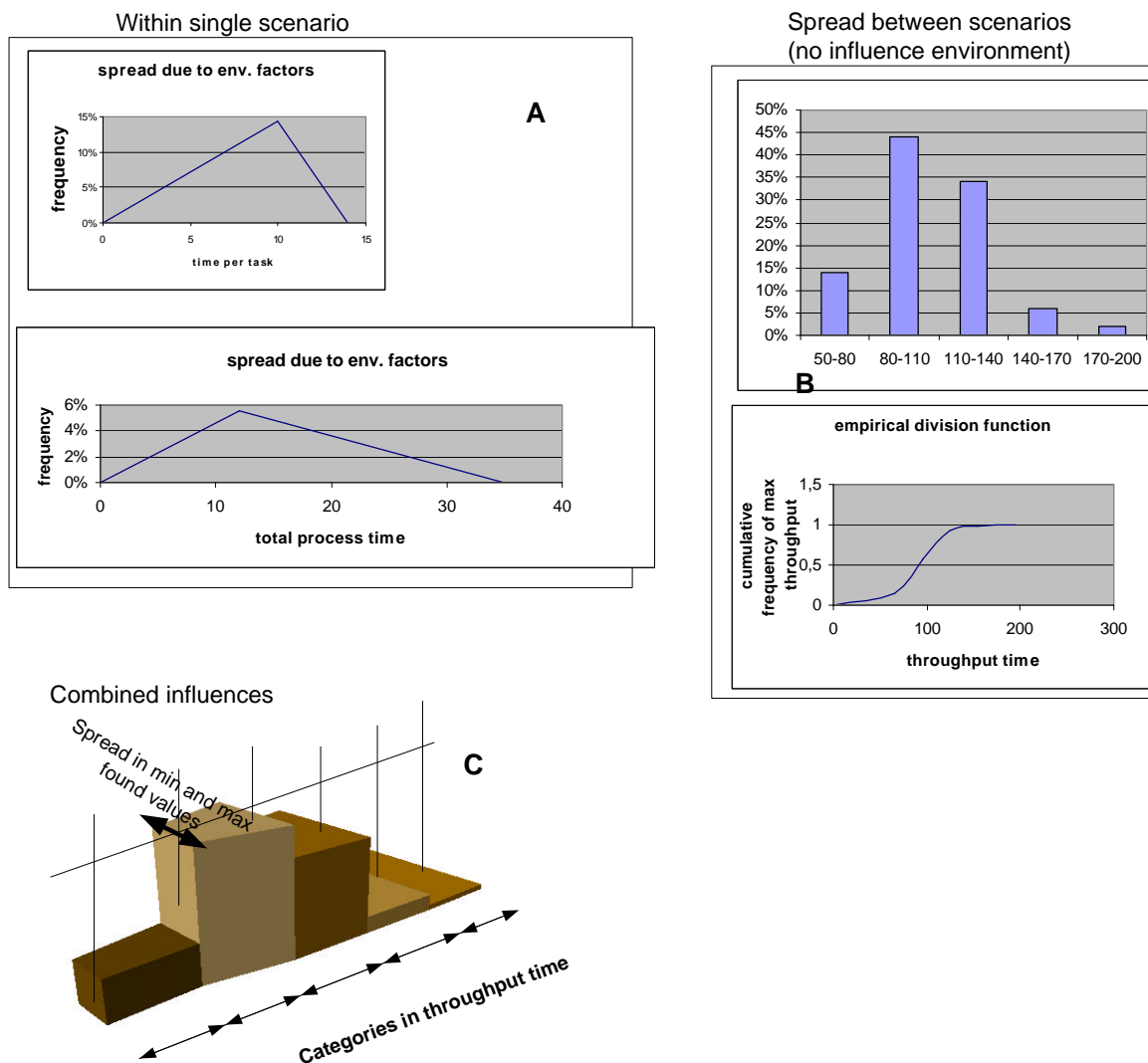


Fig.8: Illustrating variances due to delay times and task probabilities

For finding the dependency of requested capacity or delay time on product-related factors, more advanced methods of data mining exist, Kirby (2002). Influences of product related factors on delay times are not naturally present in all steps in engineering, as many tasks (for instance the example of calculating the necessary capacity) are the same for each type of product and involve the same steps. However, if the product of interest has a very unconventional type of heaters, then the calculation of heat transmission could take some more time. Gathering objective index numbers for time and capacity asks for the same approach as explained for gathering probability data.

Using probability distributions in representing the range in delays of single tasks, results in a range of found throughput times per generated solution, such as the triangular distributions in figure 8A. Only considering task occurrence influences results in distributions of throughput times for all mutants of a solution, Fig.8B. Fig.8C combines the influences by illustrating the variance in found minimum and maximum throughput times within an acceptable confidence interval per scenario, within a histogram that shows all mutants of a network.

4 Results and conclusions

4.1 Modelling HVAC case and electric system case

The method described in this paper was and is used for analyzing parts of the engineering process at the yard under study. The first case was the evaluation of the HVAC system design. It has reached the stage where factors for improvements are now systematically tested with the process tool. Definitive results and recommendations are expected in summer 2004, but interesting trends were already found. A positive side-effect of this study has been that all interviews with involved engineers and experts improved general process insight, stimulated the dialogue between different parties, and caused a lot of 'hidden' knowledge to be stored in accessible descriptions. The gathering of data to describe the quantitative aspects of the process continues to be difficult, but with rough initial values and a first sensitivity analysis this work could be reduced to a smaller subset of tasks. It has resulted in a concrete action list for diminishing the negative influence of some of the environmental factors. Also more attention is now given to a systematic approach to estimating the necessary design times for certain ship types and recognizing possible problems for new ship types. While the model is new, the information found thus far is not. However, this research improved the storing of this knowledge and its transfer between planning, purchase, production departments and management.

On the basis of these results, it was decided to continue with this approach and the design of the much more complex and much more pervasive electrical systems has been chosen as the next case.

4.2 Regarding the tool and data

The Quaestor application functions properly and gives quick results. Improvements are necessary on the satellite application for mutant rating in the area of scheduling, but no major problems are expected in this field. Further-on in the research, it is intended to substitute the rating and occurrence satellites with functions within the modeler.

The next case (where two modifications of the actual process were evaluated) shows an example of the results obtained with the Process Evaluator:

- The yard does not have its own department for HVAC engineering, but contracts out all HVAC related work to a sub-contractor.
- Yard and subcontractor implement tools for real-time exchange of 3d design information.

For this illustration, the following assumptions were made with respect to the occurrence of tasks:

- The presence of 'own' field specialists influences the number of necessary iterations in systems design.
- The presence of 'own' field specialists results in extra necessary time for deliberation.
- Using real-time 3d-exchange stimulates frequent and early input from the subcontractor in the form of 3d models.
- HVAC specialists and real-time exchange decrease the number of mistakes in production drawings.

The following assumptions were made with respect to task delay times:

- The presence of HVAC specialists making their own calculations increases the total calculation time.
- Additional time is needed for communication between field specialists and subcontractors.
- Additional time is needed for making the 3D model at the yard because of direct input from sub-contractor.

Fig.9 shows some results for the evaluation of the modifications (probability of overall throughput time per category). Looking at the weighted average index, it can be concluded that:

- The presence of HVAC field specialists positively influences throughput time.
- The use of real-time 3d exchange tools may result in a less uncertain process course.

These results are valid within a broad range of expert estimations (the results are stable when experts are able to distinguish into chances between 'very high' (90-100%) and 'medium' (50 to 90%).

These findings give confidence for further practical results of the tool.

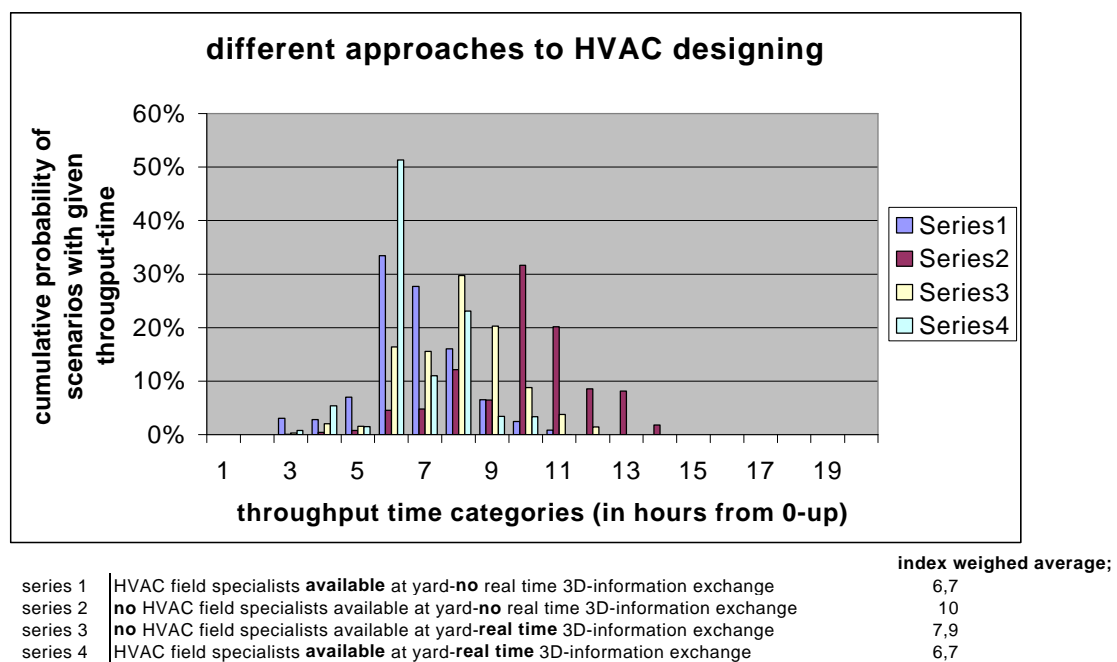


Fig.9: Preliminary results for the evaluation of the HVAC design process

References

- BOTS, P.W.G. (1989), *An environment to support problem solving*; Delft University press
- CHO, S.H.; EPPINGER, S.D.(2001), *Product development: process modelling using advanced simulation*, DETC '01, Pittsburgh
- COENEN, J.M.G.; NIENHUIS, U. (2004) *Knowledge based quantitative prognoses of the lead times of ship design processes*, Tools and Methods for Concurrent Engineering 2004, Lausanne
- COOKE, R.M. (1991), *Experts in uncertainty; opinion and subjective probability in science*, Oxford University Press
- FELDHUSEN, J.; KOY, M. (2002) *Methode zur Produktivitätsmessung für Entwicklung und Konstruktion*, Konstruktion 9 (in German), pp.49-54

- GIASSI, A. (2003), *Multidisciplinary design optimization and robust design approaches applied to the concurrent design*, COMPIT '03, Hamburg
- HALPIN, T (2001), *Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design*, Morgan Kaufman
- HEES, M.TH, VAN (2003), *Knowledge-based Computational Model Assembling*, 2003 Summer Computer Simulation Conference, July 20-24, Wyndgam Montreal, Quebec, Canada
- HERMAN, J.W. (2001), *Reducing throughput time during product design*, University of Maryland
- JOHNSON, E.W. (1996), *Analysis and refinement of iterative design processes*, dissertation
- KIRKBY, R. (2002), *Weka explorer User guide*, Univ. of Waikato
- NIEUWENHUIS, J.J.; NIENHUIS, U. (2004), *Knowledge and data reuse in ship design and engineering*, COMPIT '04, Siguenza
- PINEDO, M., (1995), *Scheduling-theory, algorithms and systems*, Prentice-Hall
- SAURER, J. (?), *Knowledge-based design of scheduling sytems*, University of Oldenburg,
- SMITH, R.P.; EPPINGER S.D. (1997), *A predictive model of sequential iteration in engineering design*, management Science, vol 43, no 8, pg 1104
- STIDSEN, T. (1996), *Job shop scheduling in a shipyard*, University of Aarhus.
- TIMMERMANS, VAN DER HEIJDEN (1983), *Oordelen en ruimtelijk keuzegedrag: formulering van beslissingsregels met behulp van logit-modellen*, from *Waar harde data ontbreken*, eds. Op 't Veld (in Dutch)
- TODD, D., (?), *Multiple criteria scheduling using genetic algorithms in a shipyard environment*, University of Newcastle
- VEEKE, H. (2002), *A simulation architecture for complex design projects*, ESS 2002, Dresden
- VERBRAECK, A. (1991) *Developing an Adaptive Scheduling Support Environment*, Dissertation Delft University of Technology
- VERBRAECK, A. (2003) Material for course *discrete modelling*, Delft University of Technology, section of systems engineering
- VRIES, D, DE (2004) *Risk Analysis for concurrent engineering: the HVAC case*, M.Sc. thesis, TU Delft, Ship Production, Mechanical Engineering and Marine Technology
- VROOM, R. (2001) *Zicht op product- en procesontwikkelingsinformatie* (in Dutch), Delft University Press, chapter 2
- WULLINK, G.; GIEBELS, M.M.T.; KALS, H.J.J.; (2002), *A system architecture for holonic manufacturing planning and control (EtoPlan)*, Robotics and Computer Integrated Manufacturing 18, pp.313-318

ODIGO - Modelling and Simulating Crowd Movement onboard Ships

Jean-Yves Pradillon, Principia Marine - OpenCasCade, Nantes/France,
j-yves.pradillon@principia.fr

Abstract

Cruise ships are becoming larger and larger and boarding more and more passengers. Both shipyard and ship owner have to deal with the crowd movement onboard to optimise the ship for emergency procedures and for operation as well. ODIGO is an original and powerful software tool based on the multi-agent method that provides the user with comprehensive and integrated functions to simulate crowd movement onboard ships. It includes pre-processing, simulation and post-processing as a whole. The paper reviews the techniques used within ODIGO to model the ship geometry, the population and the event-based scenarios. It demonstrates that a complete simulation for a large modern passenger ship requires less than 15 working days to be achieved. Human behaviour simulation tools are subject to uncertainties. For such software tools, the validation is always of a main concern. An IMO directive, MSC 1033, gives benchmarks intended to validate such tools. Results of this validation for ODIGO are presented. To highlight the capabilities of ODIGO, a real industrial application of ship evacuation simulation is demonstrated. The paper concludes with the possible developments that could be added to ODIGO in a close future.

1. Introduction

Ship design techniques have increased their efficiency to face regulations that have drastically increased their requirements in the last years. One can say that present ships are more reliable than ever. Nevertheless, the zero risk does not exist. This is why the IMO issued a new directive (MSC 1033) focusing on the evacuation process to be applied to various kind of ships and mainly for passenger ships (passenger ferries and cruise liners). This directive recommends an analysis to assess the evacuation efficiency of the ship but lets the opportunity to use either a formulae-based approach or a simulation-based approach. This later one is more difficult to achieve (need for a dedicated piece of software) but is more flexible and allows using the models for other purposes.

This is why a French national R&D project (Chantiers de l'Atlantique, Bureau Veritas, French Ministry of Industry and Principia Marine) was launched during year 2000. The purpose was to develop a mock-up software tool to demonstrate the possibilities offered by simulation to assess crowd movements onboard especially during evacuation. This project was followed by a pre-industrial project funded by Chantiers de l'Atlantique. The result of these two projects is the so called tool ODIGO. The major functions of ODIGO are presented in this paper.

The methods used in ODIGO allow using it for various other kinds of simulation as it is presented at the end of this paper. This supposes to add new objects and functions that the ODIGO architecture has been designed to support. In addition ODIGO needs some Graphical User Interface (GUI) improvements before being made available for sale. This is why Principia Marine is preparing the last development project on ODIGO to deliver the first commercial version by the end of this year.

2. The ODIGO software tool

2.1 Overview

ODIGO (from ancient Greek: I guide you) has been designed as a stand alone application for the MS Windows environment. It allows using electronic files to define geometry, it is based on the IMO recommendations to define populations and it includes post-processing facilities.

The simulation engine of ODIGO is original and has been designed from scratch in the R&D projects

already mentioned. It is based on the multi-agent method and allows carrying out simulations on a real time basis for large vessels (up to 5000 passengers + crew) on regular workstations (2 GHz processor, 512 Mb RAM). It is focused on quantitative and qualitative results assessment based on an exact geometry approach but does not include advanced display functions such as Virtual Reality animation.

ODIGO allows modelling in a quite fast way that makes it compliant with early design phases of the ship.

2.2 Pre-processing

The first step, to build a simulation model, is to create a geometry within which the agents (passengers and crew) will evolve. As far as, in early design phases of a ship project, a full 3D model is not always available, ODIGO uses a model of horizontal 2D areas located in space on a level (decks and intermediate levels) and connected one to the others by non horizontal 2D areas (stairs). The resulting model is many 2D planes represented in a 3D space: the so-called 2.5D model. The walls on a level are just represented by their basis line on the 2D surface and areas are connected together by special elements (the passages). Only public spaces are represented as far as we are not interested of what is going on in private areas such as cabins in which the movements are meaningless for the overall simulation.

The second step is to define attributes on the model elements. These elements and associated attributes include the following:

- ✓ Areas
 - Area type (public space, restaurant/bar, show room/casino),
 - Capacity used to define the maximum number of agents located in the area when simulation starts (for instance number of seats in a restaurant),
 - Displacement penalty to represent obstacles (density of tables and chairs in a restaurant for instance),
- ✓ Passages
 - The two connected areas,
 - Possible crossing cost (representing doors),
- ✓ Cabins
 - Localisation on a corridor wall representing the centre of the access door,
 - Capacity (maximum number of passengers/crew),
- ✓ Safe crafts
 - Localisation on the embarkation deck by the centre of the access point,
 - Capacity (maximum number of passengers/crew),
- ✓ Muster points
 - Localisation on the embarkation deck,

The third step is to describe the evacuation process. For that, the user must first associate cabins to one muster point (to represent where corresponding passengers/crew are supposed to muster before embarking), then safe crafts to muster points. The association of a single agent to a craft is randomly assumed by the tool from these data. The evacuation path may be driven by closing unused doors (very high crossing cost) on the way of the agents.

The fourth step allows the user describing the population characteristics. These characteristics enable the tool to locate the user at the starting position and to give them some features to describe their behaviour.

The last step before launching the simulation is to define a possible event-based scenario describing several objectives for the agents.

The experience demonstrates that an experimented user can model up to two complete decks a day for

a large cruise liner.

2.2.1 Background

The first operation to create a level is to read in the background. It is generated using a DXF file containing the general arrangement of the level. If the DXF file contains the drawing for more than a level, it is possible to create several levels from a single background file. From the DXF file, the user select the relevant part for his new level and gives corresponding information to turn it into world coordinates from the 2D coordinates of the DXF file (scale, position in space), Fig.1.

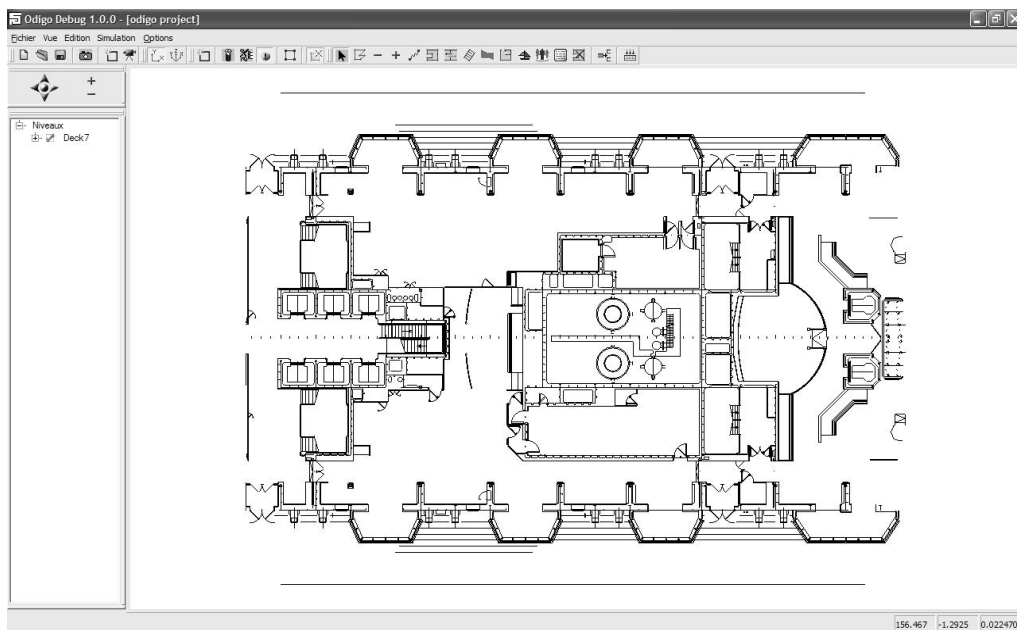


Fig.1: The background for partial deck geometry

The background is a sensitive object with a magnetic behaviour that allows the user just picking existing coordinates instead of keying in values from the keyboard.

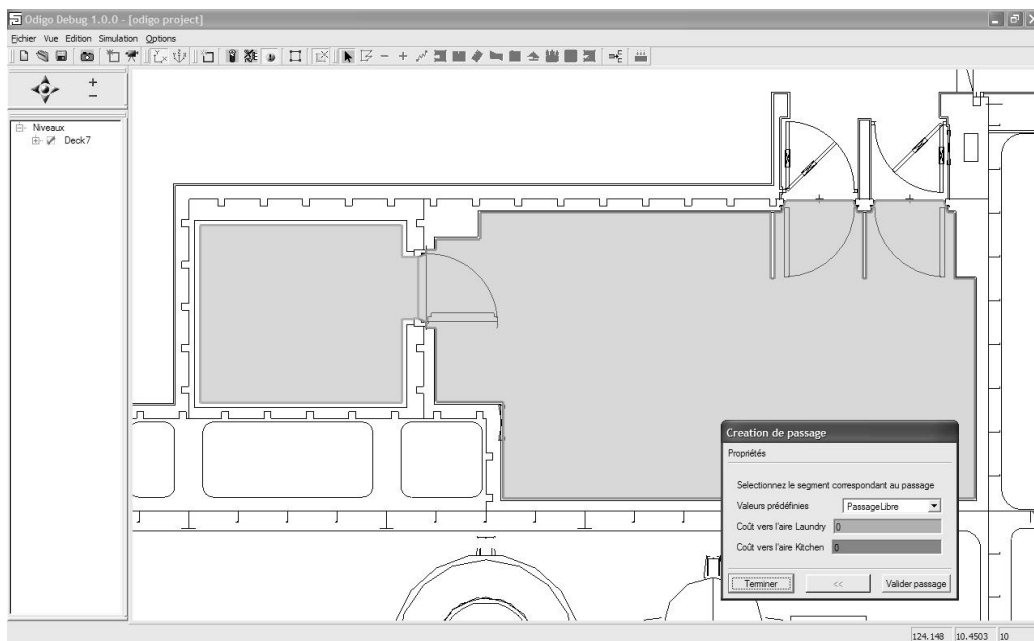


Fig.2: Defining a passage between two areas

2.2.2 Geometry

Geometry covers the previous steps two and three. To create the geometrical elements (step 2), the user has to first create polygons lying on the background. Once created, these polygons may be edited to change, suppress or add vertices. Each polygon of a level may be turned into an area that is described by a name, a type and a capacity. The default status of a polygon edge is to be regarded as a wall. Two areas sharing an edge may be connected by turning the edge from the wall to the passage status. During this operation, the costs (lost meters to cross the door) may be given for both directions (from area 1 to area 2 and from area 2 to area 1). These values may be used to model one way doors (e.g. turning wheel access), Fig.2.

Stairs are special areas created using a wizard that requires only two edges on different levels, a capacity and costs for the up and down travel directions. Resulting polygon, area and passages are automatically created, Fig.3.

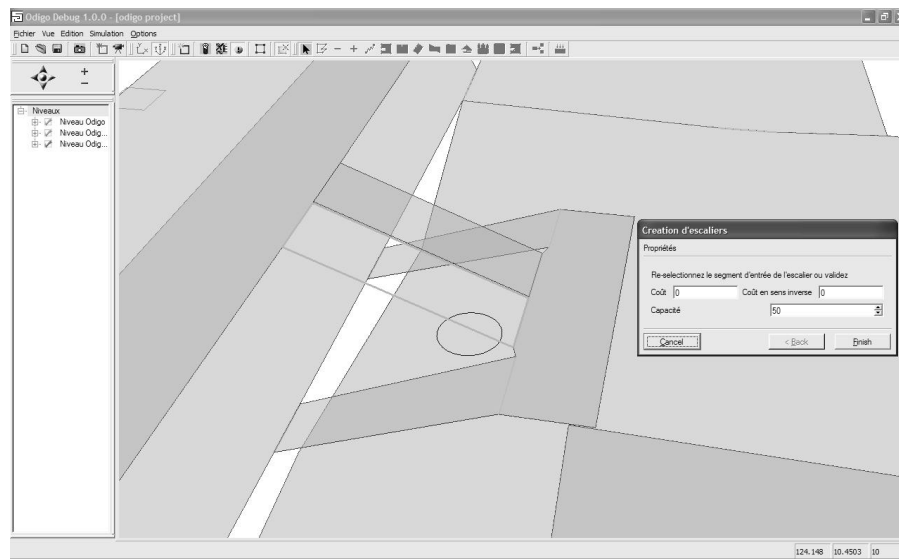


Fig.3: The stairs creation wizard

When all areas have been defined, the user must locate the cabins, the muster points and the safety crafts, Fig.4. A function is provided to check the model consistency before simulation.

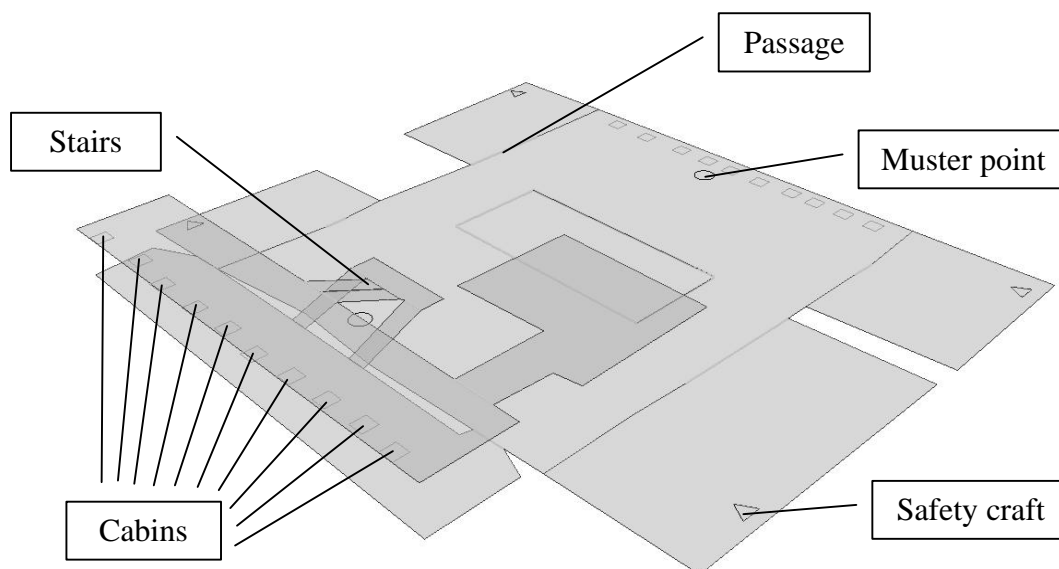


Fig.4: A sample complete geometrical model

2.2.3 Population

When many agents (passengers/crew) have to be modelled, it is almost impossible to define each agent at once. The only usable possibility is to define statistical repartitions among the population to create it. Most parameters can be defined this way and the various resulting combinations produce realistic populations with a convenient discrepancy of the characters. The statistically defined data include:

- ✓ Gender
- ✓ Age and disable ness,
- ✓ Features (speed, mind, vexation, agressivity, stress allowance),
- ✓ Reaction time,
- ✓ Initial position onboard (percentage of agents in each type of area at different times in the day).

The statistical data, Fig.5, used in ODIGO are coming from the IMO directive MSC 1033.

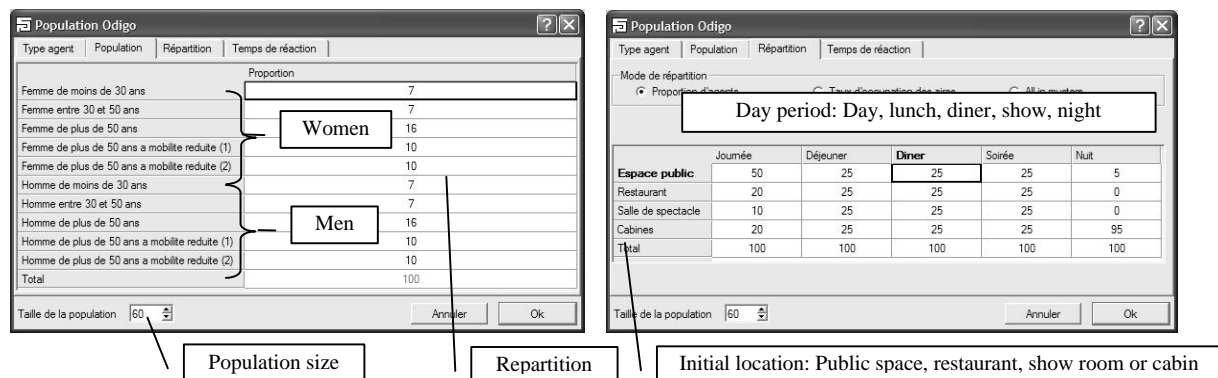


Fig.5: Age/gender and initial position repartitions

2.2.4 Scenarios

To launch a simulation, the agents must be provided with a primary destination. This is the default target that the user may choose among the traditional directions during an evacuation (cabin, muster station and craft). In case of full evacuation process simulation, the agents will receive a global plan and each part of the plan is started on an event-driven basis. To define full parameterised plans that can fit to every agent, generic events and orders have been defined.

Assuming that the evacuation process is carried out this way:

- ✓ All passengers to join cabin,
- ✓ When all passengers in cabin captain orders to join muster stations,
- ✓ When all passengers in a muster stations are mustered, then join crafts.

The corresponding scenario will be:

- ✓ All passengers' initial target is cabin,
- ✓ Event: Last passenger in cabin, Action: All join muster point,
- ✓ Event: All passengers of a muster arrived, Action: Go to crafts.

The corresponding actions of an agent will be:

- ✓ Located at the initial position,
- ✓ Wait until reaction time is elapsed to become active in the model,
- ✓ Go to cabin,
- ✓ Wait for the captain's order to leave (wait the minimum required time in cabin to put the life jacket on: 120 sec.),
- ✓ Go to muster point,
- ✓ Wait for the captain's order to embark,
- ✓ Go to safe craft.

2.3 Simulation

The simulation engine is object-oriented and it is based on a time steps approach that complies with the following schema:

- ✓ Initialisation (reading of the input files)
- ✓ Main loop
 - Time management
 - Updating external world (introduction of hazards)
 - Updating agents (realisation of actions, updating goals)
 - Dealing with conflicts from the current goals
 - Possible saving of intermediate or final results

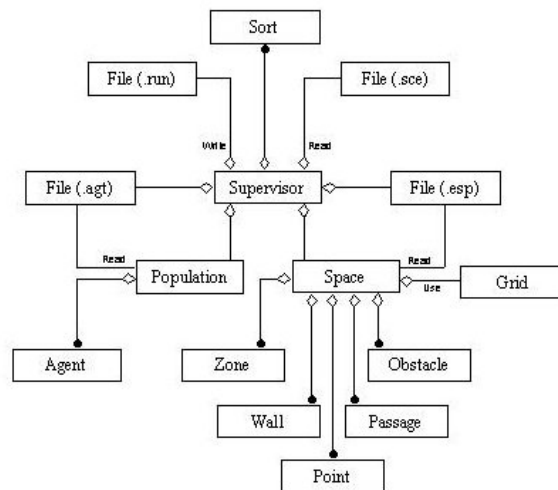


Fig.6: Schema of the relationship between Supervisor, Space and Agents

2.3.1 Input files

The input files contains the description of the scenario (time steps to use, initial position and speed of the agents, time of appearance of events such as hazard or initialisation of a specific agent), description of the space (walls, passages and zones with their status such as stairs or muster station) and description of the agents (personality such as maximum speed (physical) or maximum level of vexation (intellectual), initial state), Fig.6. These files are automatically created by the pre-processor without any user action.

2.3.2 Supervisor

The first supervisor's function is to be an interface to ODIGO database and input/output files. For instance it is in charge of reading initial data and parameters of the simulation (scenario, space, agents). Then it manages time steps (RunStep method which in turn trig the agents' RunStep method) and appearance of events as initialisation of delayed agents and hazards changing the space state (passage being blocked, fire propagation, ringing alarms). Finally, it is responsible for saving interim or final results from the simulation.

The second supervisor's function is to help agents in many aspects. For instance to deal with a negotiation with an other agent. Following the same idea, it may be mandated to give an oriented perception of the reality to the agents upon the interest of the overall population. It also includes a powerful "path finding" algorithm to provide the agents with the best way to go from one location in the model to another.

In the first implementation, the supervisor is slightly managerial (reactive agents oriented) but when the agents will get more and more intelligence, the supervisor will withdraw in the background letting more initiative to the agents (cognitive agents oriented). For instance, in case of the path finding algorithm, the managerial aspect of the supervisor is regarded as the crew role during evacuation.

2.3.3 Agents

For each time step one single agent executes the following tasks: Feeling, Thinking, and Acting. The Feeling function is carried out using a perception object communicating with the supervisor to build beliefs of the external world. The Thinking function is carried out using a planning object that builds a plan and select an immediate goal to achieve. Finally, a Goal object, selected by the planner, deals with the Acting function. To do so it runs a set of behaviour rules and combines the resulting actions to modify the internal states. We highlight some specific aspects of the schema in Fig.7.

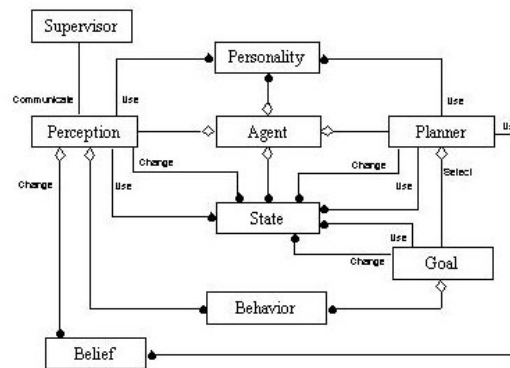


Fig.7: Schema of the relationships for the Agent object

Belief: For reacting or decide the agent needs beliefs of the external world which have not to be rational. Beliefs are maintained by the perception. Some beliefs may result from a combination of various pieces of information based on the external world and internal states. Examples of beliefs are: Presence of a passage, vicinity of other agents, obstacles on the way or states of an other agent as "lost", "panicked", "wounded"...

State: A state represents an internal feature of the agent together with its current value. At each time step a state can be updated by the perception relying upon events. Examples of states are: position, speed, tiredness, moral, stress or vexation.

Personality: To decide whether an action can be engaged the agent uses its personality. Usually, a personality is linked to a state and represents a reference value (maximum possible level for instance). The population will comprehend agent with various personalities deduced from a typical set of passengers (percentage of old people, of potential leaders...).

Goal: The agents have a stack of goals organised by priority. Some goals are immediate others are long term. For instance the father of a family may have two major goals firstly to remain grouped with his family and secondly to reach the next exit door.

2.3.4 Data for post-processing

When the simulation engine runs steps, data are recorded in a file. This set of data includes:

- ✓ Geometry (areas, passages, cabins, crafts, muster points),
- ✓ Events occurrences (gates crossed, goals reached),
- ✓ Agents' positions (every "n" time steps upon the user's settings).

As far as the simulation engine includes many stochastic parameters, two consecutive simulations will not provide with the same results. This is the reason why the user is proposed with an option to run several (50 recommended) chained simulations with the same settings. Each simulation produces output data stored in an independent file. At the end of the overall simulation these files may be exploited to draw probabilistic-like results: min values, max values, average values, standard deviation. The built-in post-processor of ODIGO is able to browse these files to present the results.

2.4 Post-processing

The post-processing functions of ODIGO include some graphic displays:

- ✓ Densities in areas,
- ✓ Routes (all routes, main routes),
- ✓ Time history of passages crossing.

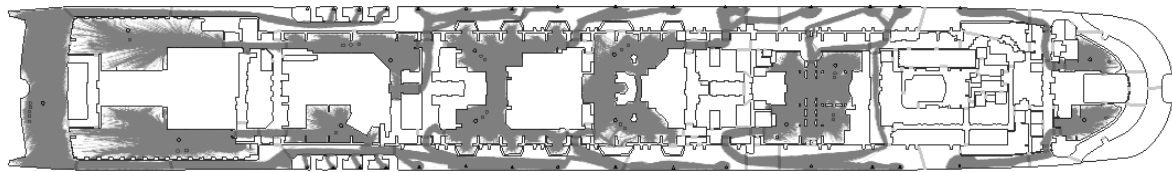


Fig.8: All routes on a cruise liner embarkation deck

If more detailed results are needed, the result files may be easily imported in spreadsheet for displaying graphs, or calculating the probabilistic results.

3. Validation

When dealing with tools involving a human behaviour simulation, a usual attitude is to wonder how the results may be validated. It is a real concern that we had to face too. We are planning drills on different kind of ships to validate some local and global behavioural models.

Nevertheless, in the present version of ODIGO, we decided to concentrate on the data provided by the MSC 1033 directive. This document includes eleven benchmark cases that are regarded as relevant, from a regulation point of view, to validate a software tool. These tests are designed to demonstrate both qualitative and quantitative features of the tool. For example the first test (the simplest one) aims at demonstrating that the tool is able to simulate that an agent walking at a speed of 1 m/s will go thru a 40 meters long corridor in 40 seconds. Others are designed to demonstrate that movements are realistic (turning at a 90° corner for instance) or that the tool handles correctly a large population with antagonistic goals, Fig.9. ODIGO passed these benchmarks and demonstrated that it is possible to use it on real industrial cases to provide data that can be accepted from a regulation point of view.

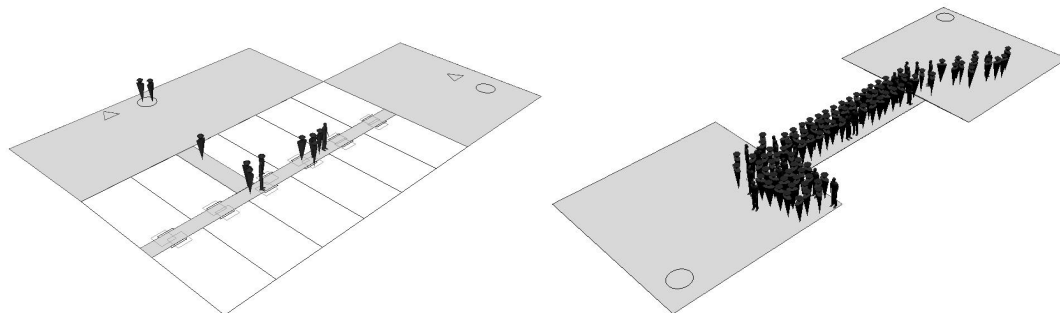


Fig.9: IMO tests: Left: Corridor movements, Right: Counter flow in a corridor

4. Industrial application

To demonstrate the features of ODIGO to both the shipyard designers and the ship owner, Chantiers de l'Atlantique decided to use it for a partial simulation of the Queen Mary 2 cruise ship. It was decided to model one typical fire zone to assess the muster time and then the full assembly deck to simulate the embarkation times. The overall evacuation process is in turn assessed by summing both times.

4.1 The QM2 fire zone muster simulation

4.1.1 Model

The selected fire zone contains all kinds of areas: Restaurants, public spaces, cabins and one large show room. The number of persons is 598 (total contents of cabins in zone). This zone also includes a muster station with two muster points (see Fig.10). As far as the simulation is disconnected from the embarkation simulation the position of safety craft is irrelevant. Two simulations have been carried out, one in day time and one during night as shown in Fig.11.

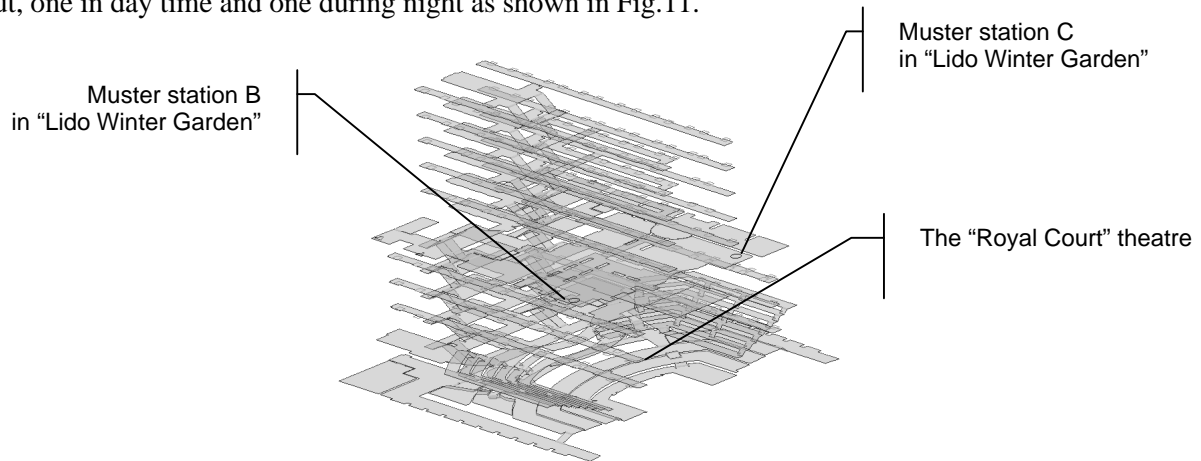


Fig.10: Fire zone 4 geometry

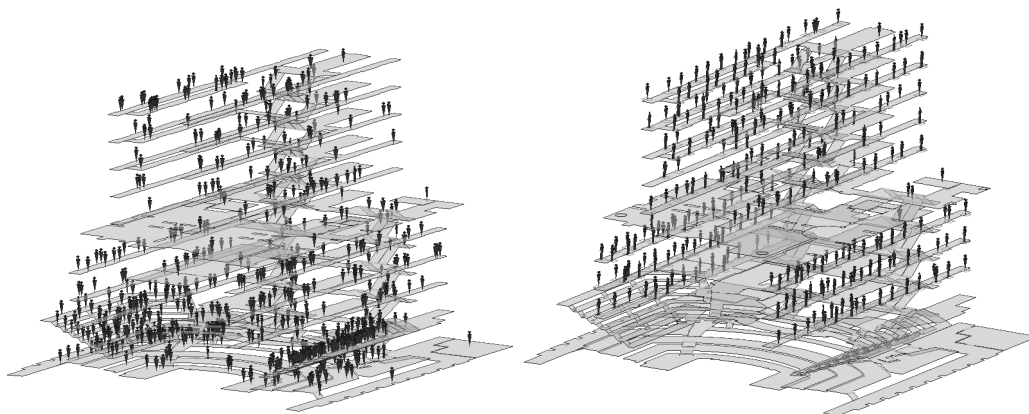


Fig.11: Start of mustering simulation in fire zone 4 (left = day, right = night)

4.1.2 Results

In Table I and

Table II, the times for "Cabins" represent the times of the last agent reaching cabin. They also represent the order time to join muster stations. Once again, all agents must stay at least 120 seconds in the cabin before leaving (picking jacket up). The times for "Muster station" represent the final times (including the time to join cabins). The dispersion time is the standard deviation of the various

time results.

Table I: Mustering times for the “Day” simulation

Target	Average	Minimum	Maximum	Dispersion
Cabins	7 mn 30 s	6 mn 40 s	9 mn 03 s	0 mn 38 s
Muster station	21 mn 22 s	15 mn 44 s	25 mn 47 s	3 mn 18 s

Table II: Mustering times for the “Night” simulation

Target	Average	Minimum	Maximum	Dispersion
Cabins	7 mn 26 s	6 mn 58 s	8 mn 08 s	0 mn 19 s
Muster station	18 mn 41 s	17 mn 23 s	20 mn 35 s	0 mn 53 s

4.1.3 Routes

Figs.12 and 13 show the routes the agents used. Fig.12 shows the more used routes. The darker the display is the more the route has been chosen. In Fig.13, all routes are displayed. The under used stairs marked on figures are explained by the fact that agents are encouraged to use the shortest way to the final target (muster station). It could be more realistic to equilibrate the two parallel stairs (as in the lower decks) but as far as this situation results in a potentially longer time it produces a security margin. Nevertheless, instead of queuing in the stairs they will have to queue at the restaurant entrance anyway.

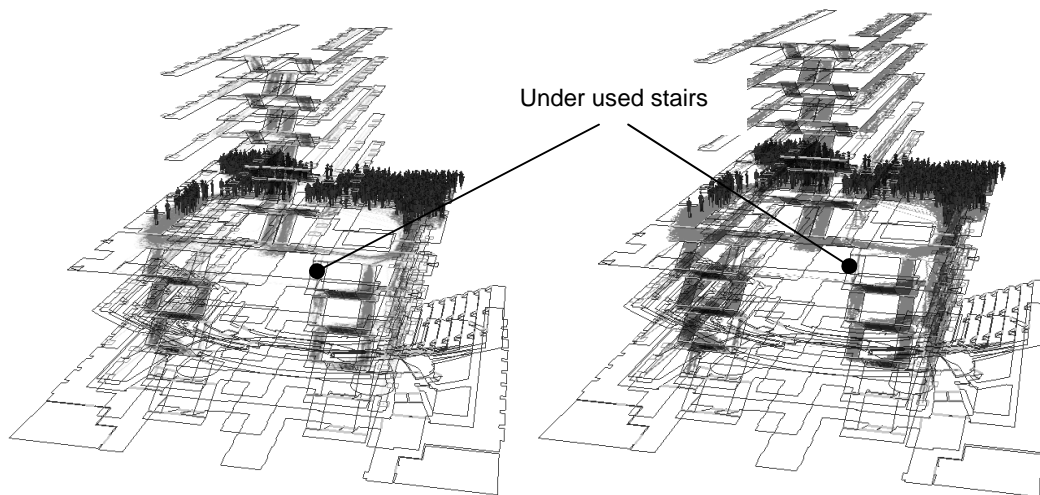


Fig.12: Mean routes – Night simulation (Left) Fig.13: All routes – Night simulation (Right)

4.2 The QM2 embarkation simulation

4.2.1 Model

The embarkation simulation supposes that all passengers are already mustered on Deck 7. This simulation regards the movements of the 4400 passenger/crew agents towards the safe crafts. Each safe craft uses a latency time (3 s) to let the passenger the time to embark. No other agent is allowed to enter the craft while the previous one is still in latency time. One single day period has been simulated as far as the agents are all located on deck 7 and aware of the process.

4.2.2 Results

Table III: Embarkation times

Average	Minimum	Maximum	Dispersion
---------	---------	---------	------------

22 mn 4 s	19 mn 22 s	26 mn 7 s	2 mn 7 s
-----------	------------	-----------	----------

4.2.3 Routes

The following figures display the routes the agents used. Fig.14 shows the more used routes. The darker the display is, the more the route has been chosen. In Fig.15, all routes are displayed. Note the alternate routes (other than recommended on the evacuation process) that agents followed. These alternate routes are marked on Fig.15.

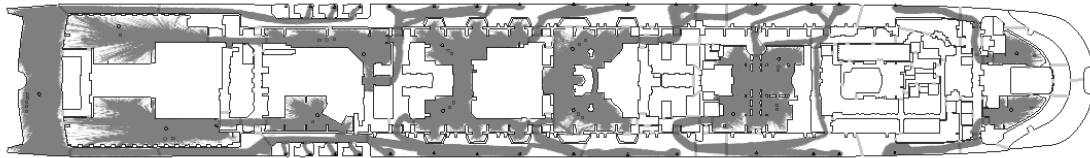


Fig.14: Mean routes

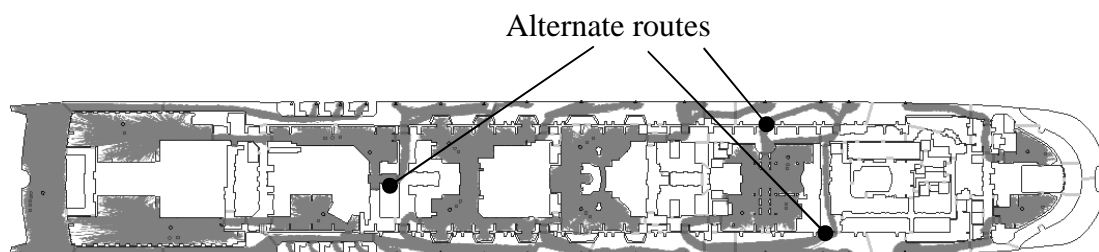


Fig.15: All routes

4.3 Summary

As a global result of all simulations, two tables are provided here after to assess the global evacuation times. Reminding the hypotheses, two sets of results are provided (one for “Day” period and one for “Night” period) that add the times of the corresponding mustering simulation to the “Day” embarkation simulation times. As far as 10 runs exist for each simulation carried out, each set of ten results has been sorted and, in combinations, the shortest time of mustering has been added to the shortest time of embarkation and so on up to the longest. This gives for the two cases (“Day” and “Night”) an ordered list from best of best to worst of worst. This adding method is not supported by a strong theory but includes the envelope cases. The normal use of ODIGO excludes this while modelling the overall structure and chaining the targets “Go cabin”, “Go muster” and “Go crafts”.

Table IV: “Day” period

Average	Minimum	Maximum	Dispersion
43 mn 26 s	35 mn 06 s	51 mn 54 s	4 mn 44 s

Table V: “Night” period

Average	Minimum	Maximum	Dispersion
40 mn 45 s	36 mn 45 s	46 mn 42 s	2 mn 59 s

The resulting times are sufficiently far from the maximum allowed evacuation time of one hour to include safety margins in the process. This also validates the evacuation routes proposed by Chantiers de l’Atlantique onboard this ship.

5. Conclusion and perspectives

The industrial study demonstrates the ability to assess evacuation times on board Queen Mary II by

simulation means. The required time to carry out such a simulation is less than ten working days. A simulation to be carried out on the overall ship is estimated at about fifteen working days. These times do not include post-processing efforts to carefully check all results. Post-processing activities on a full model may be estimated at about five working days.

The major task in the simulation is to model the ship geometry. When this task is achieved, each simulation runs almost automatically and requires a run time equal to the simulated time on a modern workstation. This allows using this tool to improve the evacuation process while optimising various evacuation data:

- ✓ Modifying routes,
- ✓ Changing doors width,
- ✓ Preventing a stair case to end right in a high flow corridor,
- ✓ Re-affecting cabin-to-muster and muster-to-craft affectations,
- ✓ Trying out various populations,
- ✓ Easily trying out more day periods than required by rules (i.e. lunch time, show time...),
- ✓ ...

All required functions to assess an evacuation process are present within ODIGO. But these functions may also be used for other purposes. For instance, for ship operation, it is possible to simulate passengers flows onboard during disembarkation (to minimise it) or at the end of a show (to empty the room as fast as possible) or in queues to access restaurants (to efficiently locate the shops or casino games along the queues). These new functions are very useful for the owner and may become new services that the shipyard may offer to customers.

References

IMO (2003), *MSC 1033*

PRADILLON, J.Y. (2003), *ODIGO: A crowd movements' simulation tool*, 2nd COMPIT, Hamburg

PRADILLON, J.Y.; TIEMS (2001), *SPECS: An innovative mustering software tool*

UTHLANDE – An Extensible Platform for Hydromechanic Calculation Modules

Uwe Langbecker, Ricardo Pereira, Germanischer Lloyd, Hamburg/Germany
{lgb, rpe}@gl-group.com

Abstract

While ship designs are getting more sophisticated, available design time shortens and requirements related to safety and quality increase. To deal with this situation it is possible to increase the use of computer aided design methodology during design phase. However, it is not sufficient to solve each design task individually, but an integrated view is needed. This paper presents UTHLANDE, an integrated environment for hydromechanic calculations in the design phase. It provides a platform with core functionality to which individual modules can be added in a way that is similar to extending the capability of web browsers by so-called “plug-in”. Currently modules for hydrodynamic aspects like sea keeping, manoeuvring, and simulation of dynamic behaviour are under development. The system architecture is presented. A key issue for a tight integration is the data model that needs to be shared by several modules. However, as each module provides specialized functionality, it usually requires extensions to the common data model. It is shown how a hybrid data model can be used to cope with both aspects. Finally, it will be demonstrated how state-of-the-art technology is used to provide a rich graphical user interface with input validation, user assistance and various post-processing capabilities.

1. Introduction

While today’s ship designs are getting more and more sophisticated, available design time shortens and requirements related to safety and quality increase. To deal with that situation, Computer Aided Design (CAD) is increasingly used during design phase. In particular, number of well-proven, advanced applications is available and used for hydromechanic design and optimisation tasks.

Hydromechanic applications have been developed over the last years and sometimes decades mostly using FORTRAN as this was the state-of-the-art programming for scientific calculations for a long time. FORTRAN was chosen also due to performance reasons and is still in use today. For each hydromechanic aspect one or more applications are available. These are chosen and executed, sometimes in a particular order. Different applications require different sets of input data in a variety of file formats. Often the results need to be post-processed for analysis, evaluation and visualization. However, various design aspects must be considered simultaneously in order to get an integrated view. It is not sufficient to run independent applications solving individual design tasks, but usually a number of iterations are needed to get a design that meets all requirements and is optimal according to certain criteria.

This led to the development of an integrated application environment for hydromechanic analysis during early design phase, named UTHLANDE¹, Pereira and Schumann (2003). This environment does not only integrate a number of applications by providing a common user interface and process control, but it also serves as a platform for further modules extending current functionality in a way that is similar to extending the capability of web browsers by so-called “plug-in”.

Initially developed by Blohm+Voss, MTG and HSVA, a joint industrial cooperation was founded in October 2003 in order to continue and extend the development. This cooperation currently consists of 9 partners from Germany including shipyards, model basins, a class society, academic institutes, and

¹ *Uthlande* (outer lands) has been used over centuries to denote a coastal area in northern Germany. It is surrounded by and directly exposed to the North Sea.

design consultants. Starting from a functional prototype, major efforts are invested to get source code that is solid, robust and maintainable.

This paper reports the current status, underlying concepts as well as work in progress. While the next section describes specific user requirements to the system, section 3 outlines the chosen approach, explaining some design principles in more detail. Section 4 focuses on implementation aspects like module testing and visualisation. Finally, some conclusions are drawn and future developments are summarized.

2. Requirements

The main goal was to develop an integrated environment for hydromechanic analysis based on a number of existing applications. It should be possible to invoke these application as well as other external standard software, e.g. for visualisation purposes. Secondly, it was crucial to provide a good graphical user interface (GUI) satisfying the needs of a ship designer. As part of such an environment the GUI should support data input, process control and result analysis. The third major work item was to establish a common data model as a basis for a tight integration between the various hydromechanic applications. It is worth mentioning that most transformation steps can be applied even if the source code is not available for a certain application.

2.1 Integration of Hydromechanic Applications

Many hydromechanic applications have been developed in FORTRAN over years and sometimes decades are available today. All of them are standalone applications mostly without graphical user interfaces. This includes the following applications

- STRIP for linear strip method for sea keeping calculations according to short-term and long-term statistics, *Söding (1994)*
- SIMBEL for non- linear strip method for sea keeping calculations according to short-term statistics and long term statistics .(stability criteria, Stockholm Agreement)*Pereira (1995)*
- SIMBEL for non-linear manoeuvring calculations
- Wave resistance prognosis / prediction with ?-Shallo
- Slow-speed Manoeuvring
- Rudder calculations – panel method
- Roll damping prognosis
- Propeller calculations based on a Lifting-Surface-Method.

It would be an enormous waste of resources (time, knowledge and hence money) to reengineer them completely. Instead they are transformed into components with moderate effort in the following way. The current file interfaces for input data and output results are turned into classes implementing the Java *Reader* and *Writer* interface, respectively. Applications, expecting predefined names for files and directories, are changed to allow variable names if the source code is available. Also, program control and error logging needed to be considered. Program control is typically implemented via command line options and/or command files. Command files and log files are treated in the same way as input data and result data, respectively. Additionally, a common mechanism for invocation and monitoring of applications was introduced; see section 3.4 for details. The whole procedure of turning legacy applications into modules is known as *wrapping*.

Hydromechanic calculations can be time consuming, particularly dynamic calculations may take up to several hours execution time. Checking whether input data fulfil certain pre-conditions before invoking a particular calculation helps to increase efficiency. Legacy applications do not always have this functionality. But even when they do it is difficult to provide feedback other than a log file. Hence, additional functionality is introduced for doing pre-condition checks and report results.

2.2 Graphical User Interface

Graphical user interfaces are state-of-the-art for home and office applications. They provide advanced input techniques for data, support visual cross checks prior to calculation, e.g., in form of diagrams as well as for post-processing and analysis of results. As most calculation modules were originally developed for batch processing, they typically lack this support. Therefore, a common GUI was needed for all calculation modules, enabling the user to run multiple calculations on the same set of input data essentially provided only once. Input data shall be stored and validated on demand, at least when a calculation is launched. Good usability shall be achieved by using identical layout, look & feel as well as by reusing input components across various applications, see Fig. 1.

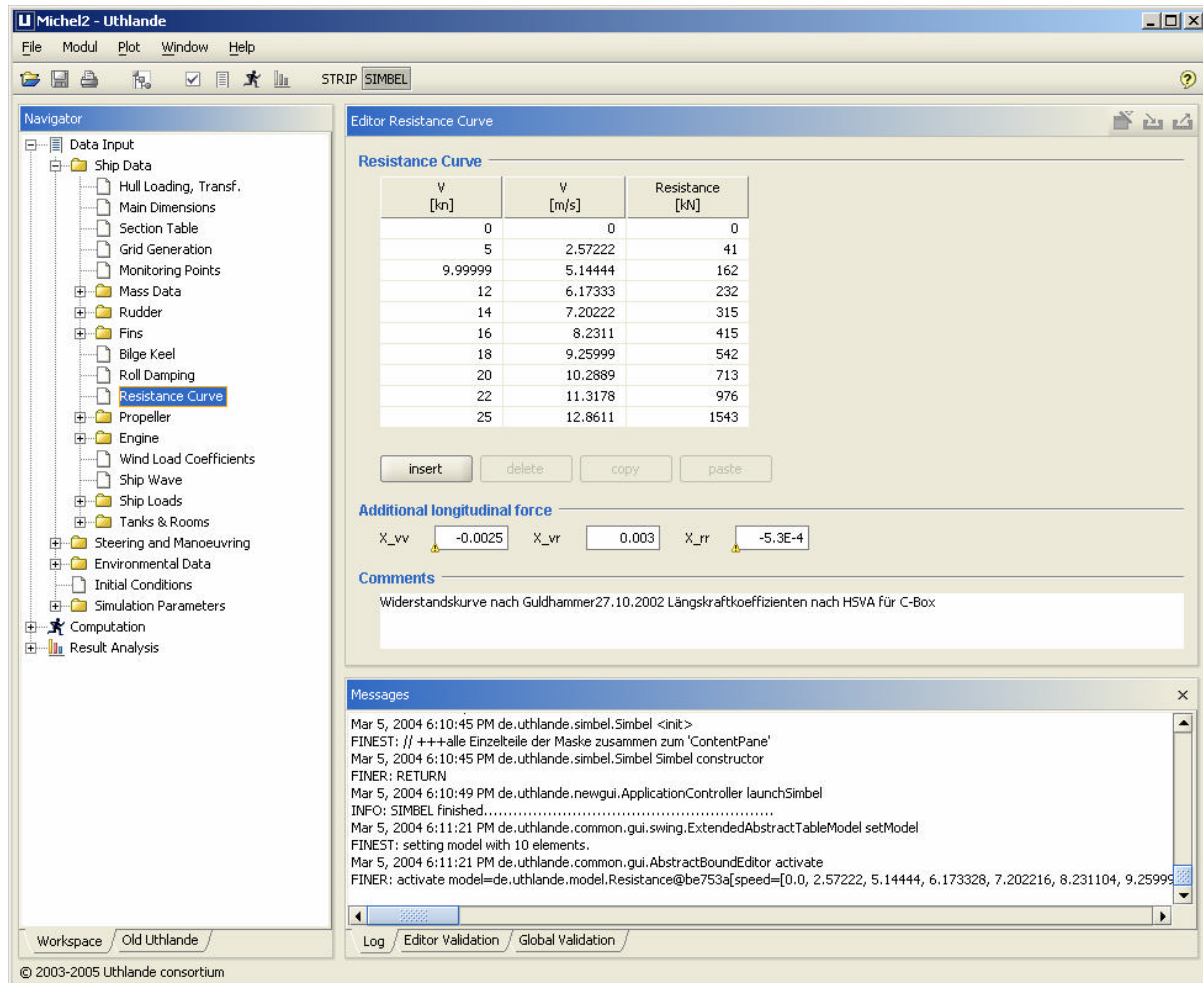


Fig. 1: UTHLANDE: Graphical User Interface including Navigator, Editor and Message window

The following components should be part of the GUI to be developed:

- Input wizards
- Editor validation
- Pre-condition checks
- Quick switch between applications

The first three items in this list help to ensure consistent input data and to reduce input errors. Input wizards assist the user by reducing the need for manual inputs and by providing good guesses for default values. Editor validation checks the format and pattern of input data. It can also validate numerical values against a predefined range of values. Editor validation is done dynamically for the active panel. Checks are performed as the user types in the data. The result is a list of problems with

different severity, e.g. warning or error. All problems relate to the active panel, i.e. they can be fixed directly. This behaviour has been adopted from integrated programming environments. Pre-condition checks are done to detect missing or wrong input prior to calculation, see previous section. They are application-specific, since a parameter required for one application, may be optional for others. Finally, switching between perspectives according to certain applications shall be possible by clicking a single button.

2.3 Import and Export of Data

Often, data sets describing the hull form, tanks, or compartments are available from CAD programs like NAPA, Rhino and FORAN. Different formats are used for describing the geometry in terms of lines, surfaces or panels. In order to avoid retyping these data, suitable interfaces, e.g. to DXF and IGES must be provided. Another type of interface is required when data from previous projects shall be imported, either partially or complete. Further interfaces, e.g. for mass data, will be required to improve the exchange of data between applications within a company, or even between a contractor and its subcontractors.

2.4 External Tools

A number of external tools are available to prepare and visualise 2D-plots, diagrams and 3D-graphics. Examples are gnuplot, and ghostscript, see *Gnuplot*, *Ghostscript*. These tools are well suited for this purpose and have been used widely for post-processing and result analysis. A single mechanism is needed to configure these tools, including path names and command line parameters. Encapsulating all details specific to the operating system, it shall be possible to invoke them in a transparent way. In fact, the same mechanism should allow to launch any external application and to control its execution. File extensions can be used to associate file types and applications.

3. System Architecture

Starting from the requirements analysis above and the fact, that the integrated environment itself shall be largely independent of the operating system, Java is a natural choice as programming language. Extensions of the standard Java Runtime Environment (JRE) are needed to exploit graphic capabilities (OpenGL, Java3D). Where appropriate, third party libraries from Open Source are used for manipulation of XML data (*Castor*), handling of user interface tasks (*JGoodies*), and other common utility task *Jakarta Commons*.

The Model-View-Controller (MVC) approach, strictly separating responsibilities between classes for domain data, controls and GUI was chosen as basic principle for the system architecture. In particular, it was important that multiple views could be applied to the same domain data object. The underlying concept for the link between views, editors (a view control) and domain objects is called *data binding*. It will be explained in section 3.4 below.

3.1 Design Principles

The following design principles were agreed as a common basis for the development. Some of these principles listed above can be measured quantitatively by so-called *metrics*. Tools are available to calculate these metrics for a set of classes.

- **Simplicity** – The program code shall be comprehensible for other developers. Even if this requires more effort or causes (moderate) losses in performance, the positive impact on maintenance, exchangeability and reusability of a module is more important.
- **Openness** – Interfaces, e.g. for file handling, import of panel and grid data, and invocation of external tools need to be documented for potential use by third party developers. The data model needs to be documented particularly well as it will be reused by all other components.

- **Modularity** – Classes and packages shall be designed for a single, clearly specified use case. The number of dependencies shall be minimized.
- **Portability** – Dependencies from the operating system shall be either avoided or encapsulated and maintained in a single place. Although Linux is the primary target platform now, porting the system to Windows may become necessary soon.
- **Reusability** – Components shall be well documented for potential reuse. Instead of copy and paste, code reuse shall be enforced by extracting common code into separate methods or classes.
- **Maintainability** – Access to operating system calls shall be encapsulated. When appropriate the factories and builders shall be used for creation of objects thereby ensuring consistency. Code needs to be well documented. The length of methods and classes shall not exceed a certain limit.
- **Efficiency** – The perceived response times for user inputs shall be short. Start-up time for the system shall be minimized by, e.g., lazy initialisation of components. Due to the high cost of object creation, input components like panels shall be created only once and shall be reused.

3.2 User Interface Framework

To create a sophisticated and responsive user interface efficiently, the third party toolkit *JGoodies Swing Suite* was used, *Lentzsch (2003)*. This toolkit complements Swing by providing components and solutions to solve common user interface tasks. It includes an alternative layout system, Look&Feel (L&F) support as well as many convenience methods for creating custom components, pre-built panels, dialogs and frames. Some libraries are freely available on Windows, Linux, and other platforms. The main benefits of using this toolkit are:

- Precise Look&Feel according de-facto standard for GUI design, *Sun Microsystems (2001 a)*.
- A new layout (*Form Layout*) that makes implementation of elegant and consistent Swing user interfaces easy resulting in short and comprehensible code.
- A large number of pre-built components including panels, dialogs, and frames that can either be used directly; or can be extended, modified and integrated. Builders and factories for creation of these components support internationalization.
- Handling of individual user settings that are stored as preferences and loaded upon start time. These preferences include window sizes, colours, Look&Feel, path names, etc.
- Central management of application launching, actions, resources, dynamic help, etc. Another major task is updating of the various interface components.
- Support for data binding allows to connect views and corresponding domain objects via a controller (here: an editor) with minimum code to be developed and maintained. Views and domain classes are largely independent; the only requirement is that they are implemented as standard bean, *Sun Microsystems (2001 b)*.
- Validation of user inputs and mark-up of validation results is supported. Validation can be done for individual components as well as for complete panels. The validation results can be processed in visualized in various forms, including icons, colours, structured lists and strings.

The toolkit includes sample applications that demonstrate good practice patterns, show how to separate concerns, and contain an architecture, that scales up well to medium sized applications.

3.3 Module Integration

In order to appear as a single application environment, modules can be bundled in various ways achieving different levels of integration. To characterise those levels integration, we consider the following sequence of tasks to be performed by every module:

- Provide input data (including validation)
- Specify calculation parameters

- Execute calculation (including job control and progress monitoring)
- Analyse results: diagrams and charts, tables, comparison, 3D graphics, animation

Now, a minimum level of integration is achieved, when only module execution is controlled by the same mechanism, but user interface, data objects and results are different for each module. All modules are executed independent of each other and no data are shared. For medium level integration the mechanisms for user interface, navigation structures and module execution are shared across all modules. The modules are loosely coupled, some concepts of data objects are shared but the code is not shared, input and output formats are different. A full integration is achieved, when in addition to the previous criteria input data are shared.

Full integration can be achieved by either monolithic systems of a single software provider or by platforms with open programming interfaces (API) allowing integration of third party modules fulfilling these contracts. This approach was chosen for UTHLANDE, see Fig. 2.

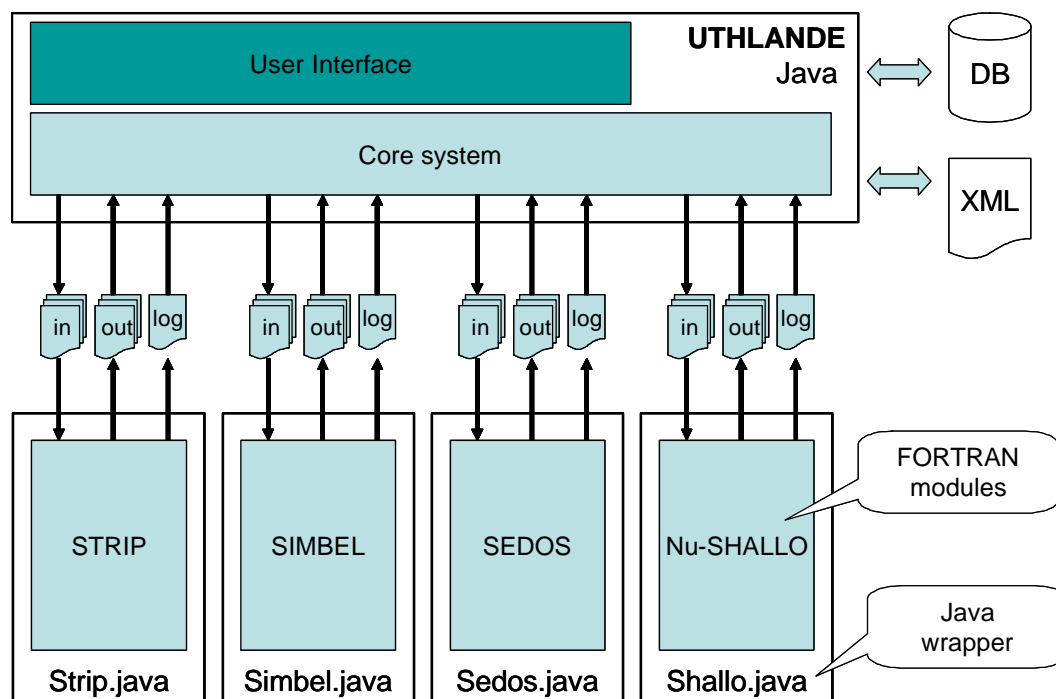


Fig.2: UTHLANDE: modular structure and interfaces

All of the modules are implemented in FORTRAN, see section 2.1. One could integrate them via the Java Native Interface (JNI). Here another option is used. The modules are wrapped, by mapping files containing input and output data from and to a single, common data model, respectively. The core system can be extended by further modules with moderate effort in the same way. Based on the common data model, different *perspectives* are defined in the user interface corresponding to individual modules. By using multiple perspectives, different aspects can be considered for the same data without the need for repeated an error prone data input.

3.4 Data Modelling

A result from the previous section it can be seen that a tight integration is only possible, if the data model is shared. To achieve this, input and output data of the various modules are analysed. Data objects are collected, compared, and harmonised where possible. This requires an in-depth analysis of concepts from the application domain – in this case from hydromechanics. This is a very essential task and it is easy to underestimate the required effort.

Harmonising data models involves introducing new abstractions and interfaces. Due to specialized functionality some properties of a concept may be needed for one, but they are not required for other calculation module. The more modules are considered, the more differences will be observed. A way to solve this dilemma is to create a core model including most common concepts and attributes. Some may be modelled as optional. All other concepts are handled in local extensions to the core model. When a concept is needed by more than one module, it will be moved into the core model, thereby extending the interface in an upward compatible way.

The root element of the data model is a project. Projects are modelled hierarchically; they consist of other domain objects like Hull, Propeller, and Rudder. Several instances of these are valid within a project to allow handling of mono-, double- and multi-hull ships. Projects are stored persistently in XML files, using an open source data binding framework for a two-way mapping between Java and XML, *Castor*, *Sosnoski (2002)*. This mapping is specified in a structured mapping file. The conversion is based on this file only without the need to generate intermediate Java classes. As XML is a standardised, hierarchically structured, clear text format that can even be edited manually, it is easy to transfer project files across machines as long as they do not contain any hardware-dependent information, hard coded path names, etc..

Within the platform code, data instances are transparently used independent of the persistence mechanism. Typically, Reader and Writer interfaces are being used allowing for file, pipe, or string representations. Later on database may need to be considered to prepare for multi-user capabilities.

In practise it is often required to handle multiple variants of a project and to compare calculation results between those. Basically, variants are considered as independent among each other, no cross references are used. However, they are tagged by a version number and a time stamp so that this association can be established when needed. If stored in a file system, a common directory is used for grouping them. Calculation results need to be stored separately and need to be associated to a set of input data. A high-level comparison between project variants regarding input data and results will be implemented.

4. Implementation

The implementation is done in Java, using JRE 1.4.2 and Java3D 1.3.1. The development is organised in a distributed way, based on commonly agreed conventions and development guidelines. A source code revision system was installed on a secure web server to manage the source code centrally.

As for most software developments, limited budget, time and resources increase the need for both fast application development and reliable delivery of software. As outlined in section 3.1, code quality has high priority. Therefore, every developer must be able to get a detailed view of the software quality anytime. For that purpose, unit testing and automated build procedures were introduced. Automated tests are being run against individual classes, modules, framework components and the complete application. This way it is possible to have a running version from the very beginning. This increases productivity, but it not only requires new tools, but also a different way of working for the team.

Based on those unit tests, code refactoring was employed to improve the design step by step, *Fowler et al. (1999)*. Only in this way it was possible to ensure that external behaviour of a hydromechanic calculation module does not change (or changes in a controlled way), while being wrapped. Also, modules can be maintained and extended internally as long as their interface does not change.

4.1 Visualisation

Scientific visualisation of technical data is important for post-processing and analysis of results from hydromechanic calculations allowing better interpretation, but also provides a means for presentation to customers, see Fig. 3. A wide range of visualisation tools and components is available to produce diagrams, 3D graphics and animations. Some of them are freely and open source including gnuplot

and ghostscript, *Gnuplot*, *Ghostscript*. A number of output formats are supported for later use in documentations and reports.

Especially for 3D graphics and animations are technology study has been done evaluating a number of Java toolkits with respect to flexibility, ease of use, performance, actuality and future perspectives, *Klemm (2004)*. The study included a comparison of Java3D, JOGL and VTK, which are all available on Windows, Linux and OS-X; see *JAVA3D*, *JOGL*, and *VTK*. Basically, all three well suited for modelling, navigation and animation of high-level 3D graphics. JOGL shows better performance and offers more low-level functionality, while Java3D and VTK provide a higher abstraction level for programming. Finally, VTK provides the largest number of algorithms and delivers best graphics quality.

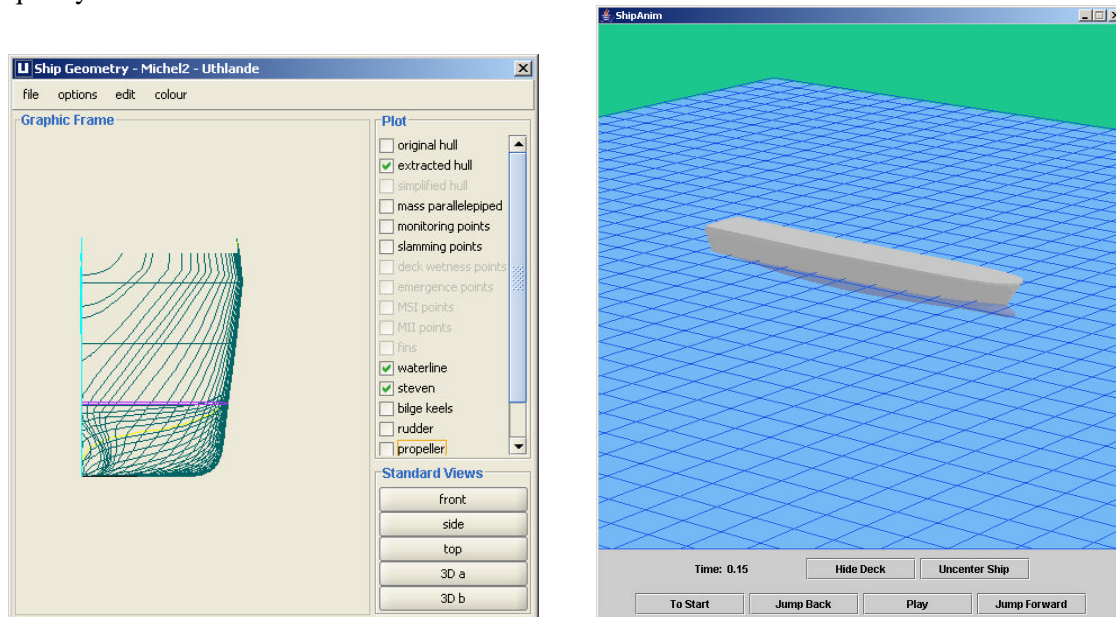


Fig.3: UTHLANDE: body lines using Java3D (left) and sea keeping animation using JOGL (right)

4.2 Process Communication

To invoke external modules in a standard way a communication protocol is needed. Here a simple protocol is proposed with the following functionality:

- Start a session
- Prepare a job (load data, specify target)
- Compose a job of individual tasks (build scripts from task)
- Run a job
- Monitor job status
- Cancel a job
- Close session

There are several options for implementing such a protocol, one of the most promising seems to be XML-SOAP. This is currently under being tested. Once the concept is fully implemented, it can be used to introduce client-server architecture. It would support multiple sessions per user, as well as multiple users working on the same server. On client side, launch configurations for existing applications need to be handled. It is useful to store, e.g. command line parameters, inputs and outputs file names. This information may even be stored persistently as user preferences between different sessions.

5. Conclusion and Future Work

The paper presents UTHLANDE, an integrated environment for hydromechanic calculations. It provides a platform with core functionality to which individual modules can be added in a structured way. It was shown how legacy FORTRAN applications are transformed into modules fulfilling a pre-defined interface.

Currently the software is developed in a joint effort by Germanischer Lloyd, Marinetechnik GmbH, HSVA, JaFo Technologie, Jos. L. Meyer GmbH, TU Hamburg-Harburg and others. Developments will continue until 2006. Until then the functionality of the system will be extended by incorporating further calculation modules, which are either available or will be developed. Deployment and update over Internet as well as a user-friendly launch environment will be addressed by introducing a JNLP based solution.

So far only single users are supported. If multi-user capability and concurrent use are required, a database solution will be considered for persistent storage of projects.

Based on the concept of process communication outlined in section 4.2, client-server architecture may be chosen, allowing for execution of time-intensive calculations on a powerful remote server, while clients can be run locally on standard desktop PC. This would have the positive side-effect, that client and server can run on different operating systems with the chance to overcome OS-dependencies of some modules.

Finally, the data model cannot be considered absolutely stable. Hence the corresponding XML schema is expected to change as well. Multiple versions are to be maintained in an upward compatible way. Thus an evolution strategy is needed, keeping track of schema changes in a controlled way.

Acknowledgements

The development of UTHLANDE is funded by a cooperation of the following partners: Aker MTW Werft GmbH, Blohm+Voss, Fr. Lürssen Werft GmbH & Co KG, Germanischer Lloyd AG, HSVA, Jos. L. Meyer GmbH, Marinetechnik GmbH, Nordseewerke GmbH, Schiffbau-Versuchsanstalt Potsdam GmbH.

The authors would like to thank Carsten Schumann, Richard Ems, Anja Siems, Jan-Helge Deutscher, and Sunay Yaldiz for many valuable discussions.

References

CASTOR; *An open source data binding framework for Java[tm]*, www.castor.org

FOWLER, M.; BECK, K.; BRANT, J.; OPDYKE, W.; ROBERTS, D. (1999), *Refactoring: Improving the design of existing code*, Addison-Wesley

GHOSTSCRIPT; *An interpreter for the PostScript language and for PDF*, <http://www.cs.wisc.edu/~ghost/>

GNUPLOT; *A command-driven interactive function plotting program*, <http://www.gnuplot.info/>

JAKARTA COMMONS; *A Jakarta subproject focused on all aspects of reusable Java components*, <http://jakarta.apache.org/commons/>

JAVA3D; <http://www.j3d.org>

JGOODIES, www.jgoodies.com

JOGL; <https://jogl.dev.java.net>

KLEMM, ST. (2004), *Vergleich und Bewertung verschiedener Visualisierungstechnologien zur Schiffs- und Seegangsanimation in einer Java-Umgebung mittels Prototyp-Implementation*, Diploma Thesis, FH Wedel, in German

LENTZSCH, K. (2003), *Erste Hilfe für Swing*, JavaMagazin, 11/2003, pp. 31-35, in German, see also www.jgoodies.com/articles/

PEREIRA, R.; SCHUMANN, C. (2003), *Programmbeschreibung UTHLANDE*, Report 205/03, JAFO Technologie, in German

PEREIRA, R. (1995), *Bewegungen und Belastungen von Ein- und Doppelrumpfschiffen im Seegang unter Berücksichtigung nichtlinearer Effekte*; MTG Marinetechnik GmbH, Report 651/7/1041-001, in German

SÖDING, H. (1994); *Beschreibung des Programms STRIP*, Technical Report, University Hamburg, in German

SOSNOSKI, D (2002); *A look at XML data binding for Java using the open source Castor project*, <http://www-106.ibm.com/developerworks/xml/library/x-bindcastor/index.html>

SUN MICROSYSTEMS (2001), *Java[tm] Bean Specification*, <http://java.sun.com/products/javabeans/docs/spec.html>

SUN MICROSYSTEMS (2001), *Java[tm] Look and Feel Design Guidelines*, 2nd edition, Addison-Wesley, see also <http://java.sun.com/products/jlf/>

VTK; <http://www.vtk.org>

Three-dimensional CFD Simulations Using a Free-Surface Capturing Strategy

Patrick Queutey, Michel Visonneau

Laboratoire de Mécanique des Fluides-UMR6598, Ecole Centrale de Nantes,
Patrick.Queutey@ec-nantes.fr, Michel.Visonneau@ec-nantes.fr

Abstract

This paper describes a modern free-surface capturing methodology implemented in an unstructured finite-volume flow solver. A systematic grid-refinement study and a comparison of several discretisation schemes for the transport equation of concentration is performed to evaluate the influence of the discretization error on the interface smearing. Detailed comparisons with experiments carried out at model scale on the Series 60 bare hull ($CB=0.60$) (Longo & Stern, 2002) illustrate the high degree of reliability of this approach, provided that adequate compressive discretisation schemes are used with a reasonable number of grid points.

1 Introduction

Traditionally, the most frequent way to evaluate hydrodynamic performances of a new ship is through tank tests. The use of CFD tools for predicting the powering performance of ships is not frequent because of the numerous physical difficulties which characterize the flow around a real ship. Among the hard points which have to be solved to get a reliable simulation of a full-scale ship flow, one can list, without being exhaustive, the accurate simulation of ship stern flows which is highly dependent on turbulence modeling, the modelisation of the hull/propeller coupling and the simulation of the free-surface deformation. On one hand, one must also notice that the accuracy of CFD tools for predicting crucial global quantities like resistance and self-propulsion factor is still limited. On the other hand, CFD is complementary to towing tank tests because it provides a large amount of detailed information on the flow which helps the designer to improve the performance of new ship. Therefore, as long as simplified configurations are considered, the large increase of computational power and memory makes now possible to have recourse to Computational Fluid Dynamics (CFD) tools which may be integrated in fully automated optimization procedures during the design phase. When the challenges related with turbulence modelisation and ship hull optimization have been treated in (Duvigneau & Visonneau, 2003) and (Deng & Visonneau, 1999), this paper is devoted to the simulation of free-surface flows and its requirements in terms of simulation strategy, discretisation schemes accuracy and computational power.

Two different methodologies are used for the representation of a viscous free-surface : the surface capturing and surface fitting strategies. In the former method, the flow is computed on a fixed grid both in the air and the water and the location of each fluid is determined by solving a transport equation for a concentration function which is equal to zero in the air and one in the water, for instance. In the latter method, the grid is fitted to the free-surface where adequate boundary conditions are written and deforms with the evolution of the solution during the computations. Although this approach is less demanding in terms of computational power because the flow is only computed in the water, one often faces severe problems of robustness because the wave-breaking can not be tolerated. The growing use of unstructured grids makes also possible the concentration of hexahedric elements near the walls and the interface to get an accurate description of the density discontinuity and tetrahedrons may be used elsewhere, which results in a significant reduction of the number of grid points and makes more affordable a surface capturing strategy. This is the reason why the free-surface capturing strategy is more and more attractive, despite its higher computational cost.

This paper is devoted to a presentation of the specific discretisation schemes which are required to get

an accurate description of a density discontinuity. An original procedure is described to avoid, as much as possible, the limitations of the time step caused by the Courant number criteria. Moreover, the influence of the number of grid points and discretization schemes is analyzed by comparing the respective numerical solutions to an extensive experimental database (Longo & Stern, 2002).

2 Flow Solver

The ISIS flow solver, developed by DMN (Division Modélisation Numérique i.e. CFD Department of the Fluid Mechanics Laboratory), uses the incompressible unsteady Reynolds-averaged Navier Stokes equations (RANSE). The solver is based on the finite volume method to build the spatial discretization of the transport equations. The face-based method is generalized to two-dimensional, rotationally-symmetric, or three-dimensional unstructured meshes for which non-overlapping control volumes are bounded by an arbitrary number of constitutive faces. The velocity field is obtained from the momentum conservation equations and the pressure field is extracted from the mass conservation constraint, or continuity equation, transformed into a pressure-equation. In the case of turbulent flows, additional transport equations for modeled variables are solved in a form similar to the momentum equations and they can be discretized and solved using the same principles. Incompressible and non-miscible flow phases are modeled through the use of conservation equations for each volume fraction of phase.

2.1 Conservation equations

The flow solver can deal with multi-phase flows and moving grids. Considering incompressible flow of viscous fluid under isothermal conditions, mass, momentum and volume fraction conservation equations can be written as (using the generalized form of Gauss' theorem):

$$\frac{\partial}{\partial t} \int_V \rho dV + \int_S \rho (\vec{U} - \vec{U}_d) \cdot \vec{n} dS = 0 \quad (1a)$$

$$\begin{aligned} \frac{\partial}{\partial t} \int_V \rho U_i dV + \int_S \rho U_i (\vec{U} - \vec{U}_d) \cdot \vec{n} dS \\ = \int_S (\tau_{ij} I_j - p I_i) \cdot \vec{n} dS + \int_V \rho g_i dV \end{aligned} \quad (1b)$$

$$\frac{\partial}{\partial t} \int_V c_i dV + \int_S c_i (\vec{U} - \vec{U}_d) \cdot \vec{n} dS = 0 \quad (1c)$$

where V is the domain of interest, or control volume, bounded by the closed surface S with a unit normal vector \vec{n} directed outward.

The effective flow physical properties (viscosity μ and density ρ) are obtained from each phase physical properties (μ_i and ρ_i) with the following constitutive relations:

$$\rho = \sum_i c_i \rho_i \quad ; \quad \mu = \sum_i c_i \mu_i \quad ; \quad 1 = \sum_i c_i \quad (2)$$

When the grid is moving, the so-called *space conservation law* must also be satisfied:

$$\frac{\partial}{\partial t} \int_V dV - \int_S \vec{U}_d \cdot \vec{n} dS = 0 \quad (3)$$

2.2 Numerical framework

2.2.1 Spatial discretization

All the flow variables are stored at geometric centers of the arbitrary shaped cells. Surface and volume integrals are evaluated according to second-order accurate approximations by using the values of integrand that prevail at the center of the face f , or cell C , and neighbor cells. The various fluxes appearing in the discretized equations are built using centered and/or upwind schemes. For example, the convective fluxes are obtained by two kinds of upwind schemes. A first scheme available in the flow solver, (HD) for Hybrid differencing, is a combination of upwind (UD) and centered (CD) schemes. Contrary to a practical approach (Demirdžić & Muzaferija, 1995; Ferziger & Perić, 1996) where CD/UD blending is fixed with a global blending factor for all faces of the mesh, the HD scheme results from a local blending factor based on the signed Peclet number at the face. An other upwind scheme which is implemented in ISIS, is the Gamma Differencing Scheme (GDS) (Jasak, 1996). Through a normalized variable diagram (NVD) analysis (Leonard, 1988), this scheme enforces local monotonicity and convection boundedness criterium (CBC) (Gaskell & Lau, 1988). A pressure equation is obtained in the spirit of the Rhie and Chow (Rhie & Chow, 1983) procedure and momentum and pressure equations are solved in an segregated way like in the well-known SIMPLE coupling procedure.

2.2.2 Discretization schemes for the transport equation of concentration

Except the convection terms and volumetric mass fluxes, inter-facial quantities q_f are rebuilt linearly from the quantities themselves and their available cell-centered gradients. In order to assume face bounded reconstructions and to avoid unrealistic oscillations, the search for an acceptable compromise between accuracy and boundedness of $0 \leq c_i \leq 1$ is a key point (Jasak *et al.*, 1999; Pržulj & Basara, 2001). Moreover, the method must also preserve the sharpness of the interface through the transport equation (1c).

Considering normalized variables, Eq. (4), the CBC criterium (Leonard, 1988) implies that, in the NVD diagram Fig. 1(a), the only scheme which unconditionally satisfies the boundedness criterium is the first-order upwind differencing scheme (UDS).

$$\tilde{q} = \frac{q - q_U}{q_D - q_U} \quad (4)$$

The second-order centered differencing scheme (CDS) is only useful in the range $0 \leq \tilde{q}_C \leq 1$. In order to provide a continuous switch from UDS to CDS and CDS to UDS, the GDS scheme illustrated into the NVD diagram by Fig. 1(b) yields a first answer to that requirement.

However, the GDS scheme is based on upwind or centered differencing and could not be appropriate for maintaining the sharpness of an interface since numerical diffusion is introduced. On the other hand, this scheme does not suffer from any Courant number limitation. An alternate treatment of numerical diffusion problem for interface capturing method is to use the donor-acceptor cell concept (Hirt & Nichols, 1981). This can be realized with the GDS scheme if one introduces downwind differencing (DDS) to build the Inter-Gamma scheme (Jasak & Weller, 1995), Fig. 2, since compressive characteristics are required for an accurate interface capturing.

The main disadvantage of the IGDS scheme is a Courant number limitation : $Co < 0.3$ in multidimensional cases. To overcome that difficulty, the following correction (5) is used which proved satisfactory in most applications (Muzaferija & Peric, 1998). Then, the adopted scheme for interface capturing MGDS, results from the Courant number limited scheme IGDS with corrections from (5).

$$\begin{aligned} \tilde{q}_f &= \tilde{q}_C + (\tilde{q}_f - \tilde{q}_C) \frac{0.7 - Co}{0.7 - 0.3} & \text{if } 0.3 \leq Co \leq 0.7 \\ \tilde{q}_f &= \tilde{q}_C & \text{if } Co \geq 0.7 \end{aligned} \quad (5)$$

2.2.3 Temporal discretization

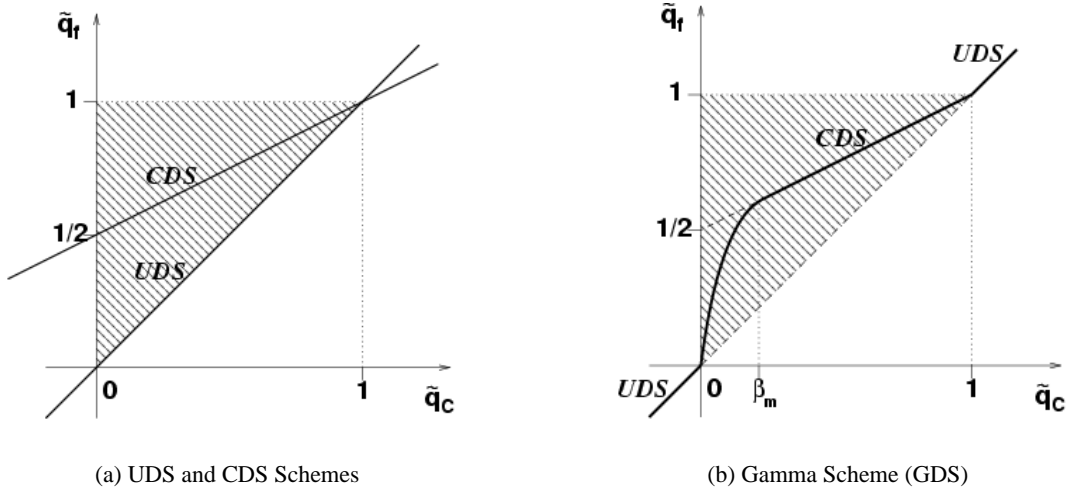


Figure 1: Numerical Schemes in NVD diagram

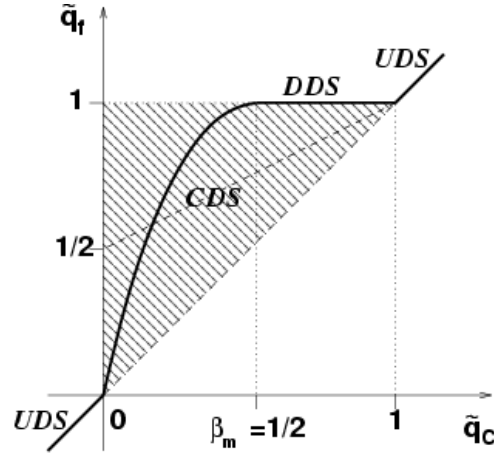


Figure 2: Inter-Gamma Differencing Scheme (IGDS)

The temporal discretization is based on a three-step upwind discretization ensuring a second-order accuracy. The convection and diffusion are treated implicitly.

2.3 Boundary conditions

Wall function conditions are imposed on the walls. Symmetry conditions are written in the vertical plane of symmetry for the computations without drift. Computations are carried out in a full three-dimensional domain for the validations with a steady drift. To speed-up the convergence towards a steady state, the computations are performed around a moving ship with a constant acceleration rate from $T=0$ to $T=1$ followed by a constant speed.

3 Validation on Series 60 bare hull in drift

3.1 Description of the test case

In order to assess the reliability of this free-surface capturing strategy, the Series 60 database (Longo & Stern, 2002) has been chosen because of the availability of several well documented drift configurations. Three drift angles (0, 5 and 10 degrees) have been retained. The computations are performed on a parallel

Grid	Number of grid points	Number of processors	Total CPU time
Coarse	6.52 10^5	16	7 hours
Medium	2.82 10^6	32	2.5 days
Fine	3.89 10^6	32	5 days

Table 1: Characteristics of the grids used in the grid independence study

computer using MPI and the $k - \omega$ SST model is selected to model the turbulent correlations. An original and robust wall-function (Deng *et al.*, 2004) boundary condition is employed on the wall instead of a near-wall low Reynolds number formulation in order to avoid, until now, any difficulty related with the behavior of the interface in the vicinity of the wall when very fine grids suited to a low Reynolds number formulation are employed.

In order to get a well converged result, especially to satisfy the wave propagation in the far field, it is mandatory to compute up to an adimensionnal time $t=40$ in the most severe test case (drift angle of ten degrees).

This is why it has been decided to start the computation with the GDS scheme which has no constraints on the Courant number, and then switch to the compressive MGDS scheme with a time step ten times smaller to restore the sharpness of the interface at the end of the computations. This procedure which is only valid for steady flows allows free-surface computations for reasonable CPU cost. All the presented numerical simulations have been performed on a SGI ORIGIN 3800 R14000/500Mhz machine.

3.2 Grid independence study

To evaluate the influence of the number of discretization points on the solution, a grid independence study has been carried out on three unstructured grids whose topology is described in Fig. 3 with only some significant domains. To avoid too many lengthy computations, this study is limited to the drift angle of 10 degrees. In order to avoid the smearing of the interface, hexahedric finite volumes are used near the interface and tetrahedric elements are preferred in the far field to reduce the number of points. Hexahedric elements are also retained near the wall to guarantee an accurate discretisation of the viscous diffusion effects.

The characteristics of the three grids employed are described in the table 1 with CPU time spent. Figs. 4 show the free-surface elevation for a steady drift of 10 degrees computed on the afore-mentioned three grids with the MGDS discretisation scheme. From these figures, one can notice that the coarse grid is definitely too coarse since it is not able to capture the correct Kelvin angle although the main divergent waves are already captured. The medium and fine grids reveal far more accurate representations of the free-surface with a correct Kelvin angle, steeper divergent and transversal waves and, for the fine grid, many secondary divergent waves in good agreement with the experiments as it will be illustrated in the next sections.

3.3 Influence of the discretisation scheme

The challenge posed by the discretisation of a transport equation for the concentration is the accurate modeling of a contact discontinuity, i.e. the free-surface. From the previous section, it appears that the compressivity is theoretically crucial to keep the interface as sharp as possible, even if this property is satisfied through a severe Courant number limitation. It is therefore interesting to evaluate on fine grids and real-life problems, the influence of the discretisation error to check if it is realistic to use non-compressive discretisation schemes which result in faster computations since no criteria has to be satisfied on the Courant number. Figs. 5 show comparisons of the wave profiles on the wall of a Serie 60 hull obtained at three different drift angles, namely 0, 5 and 10 degrees on the fine grid.

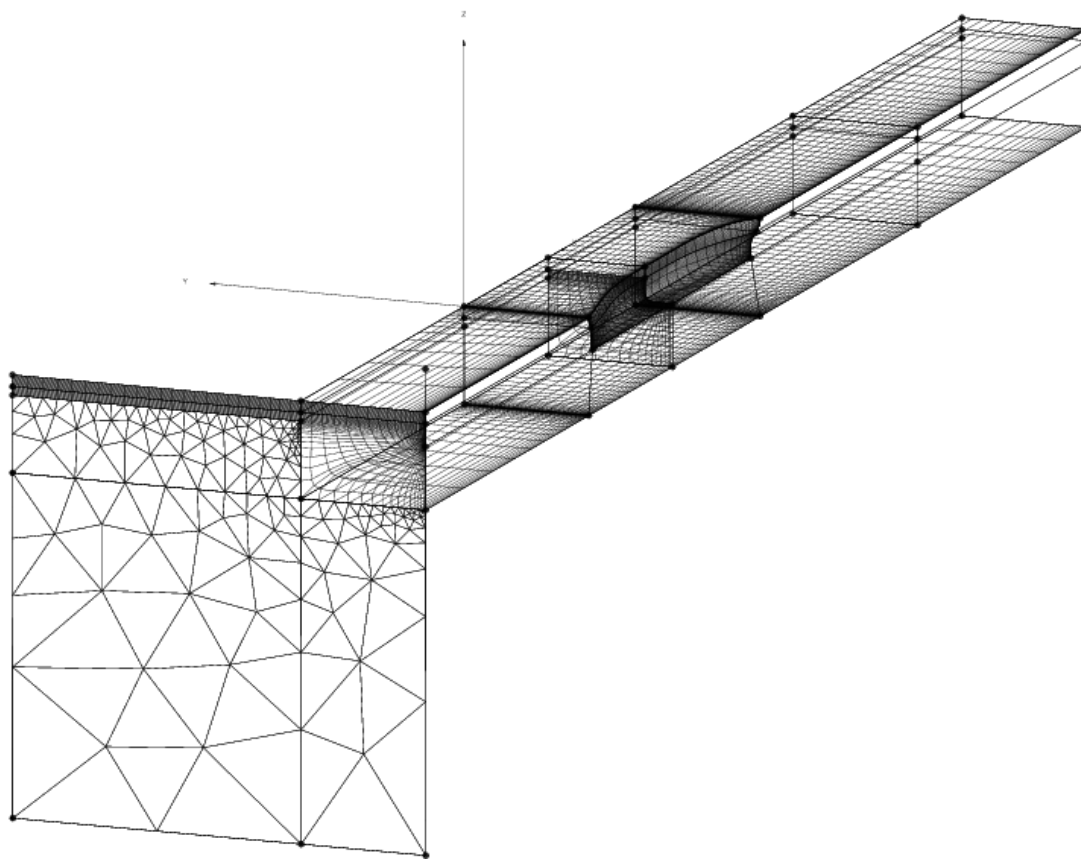


Figure 3: Typical unstructured grid topology for a free-surface flow

>From these figures, one may conclude that the discretisation schemes play a minor role since the results are quite similar. However, if one considers Figs.6 which compare the free-surface elevation field all around the ship hull, it is clear that the discretisation schemes still play a central role even if these results are obtained on the finest grid. Not only the amplitudes of the main divergent waves are modified but one can observe that the free-surface field computed with MGDS contain many secondary waves inside the Kelvin angle in good agreement with the experiments, a behavior which is not simulated by the cheaper GDS scheme.

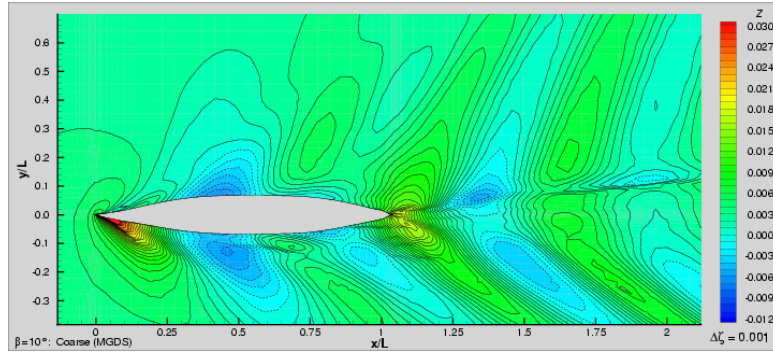
Particularly, it is noteworthy to mention that, at 5 and 10 degrees drift angles, the bow wave spray sheet detaching from the hull and merging with the downstream breaking bow wave, is remarkably captured by the MGDS simulation.

Additional illustrations of the influence of the compressive property will be given in the section relative to wave cuts which will confirm this analysis.

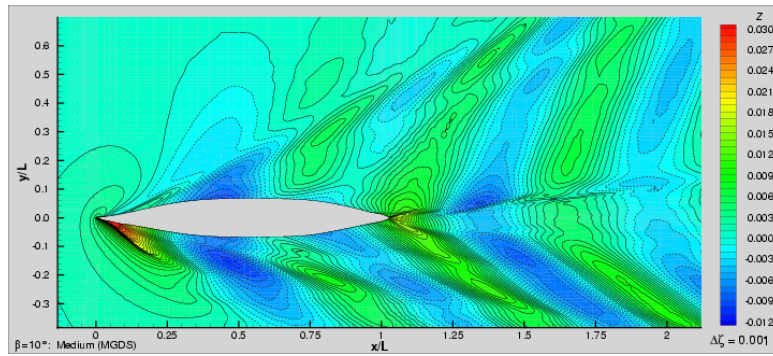
3.4 Comparisons to the experiments

3.4.1 Free-surface elevation

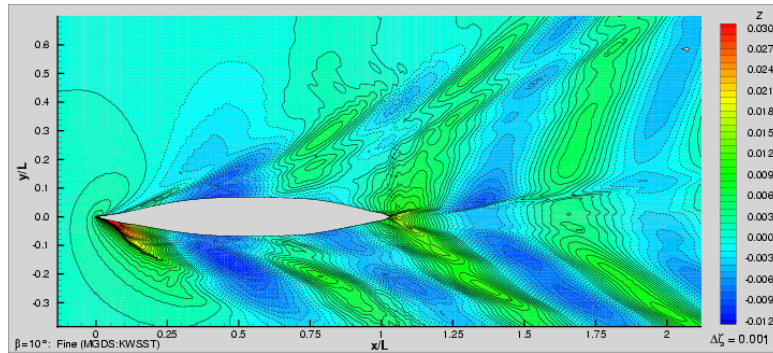
Fig. 7 shows the computed free-surface elevations compared to the experiments for a drift angle of 0 degree. These results like those relative to higher drift angles are obtained on the fine grid. One can notice the very good agreement between computations and experiments since all the divergent waves are accurately captured by the simulation. The transverse waves at the stern of the ship are also in very good agreement with the experiments.



(a) Coarse grid



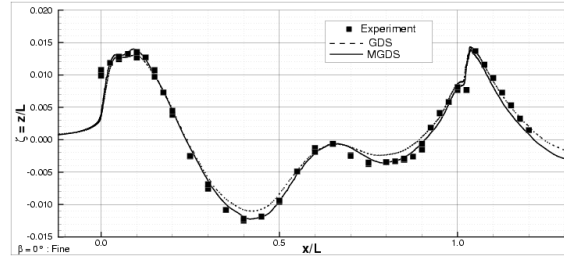
(b) Medium grid



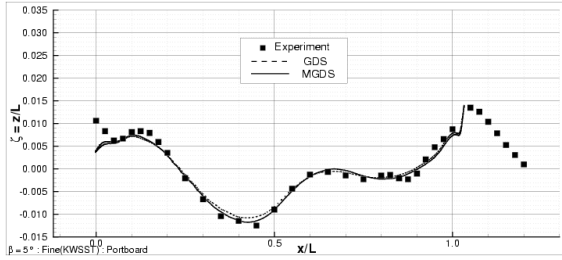
(c) Fine grid

Figure 4: Series 60 with a drift angle of 10 degrees. Influence of the number of grid points

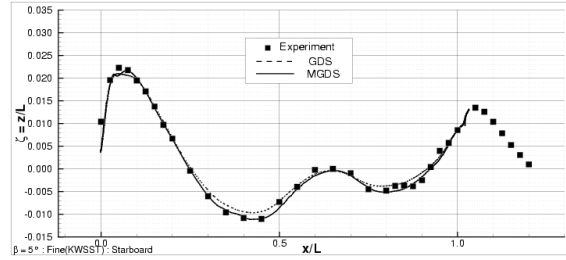
Fig. 8 shows the computed free-surface elevations compared to the experiments for a drift angle of 5 degrees. Lastly, Fig. 9 shows the same comparison for a drift angle of 10 degrees. The wave contours are significantly influenced by the drift angle since the wave amplitudes are drastically increased/decreased on the windward/leeward sides. It appears that the influence of the compressive MGDS scheme is far more important on the windward side because the water is pushed away from the hull resulting in a downwind dependency, only accounted for by the compressive scheme.



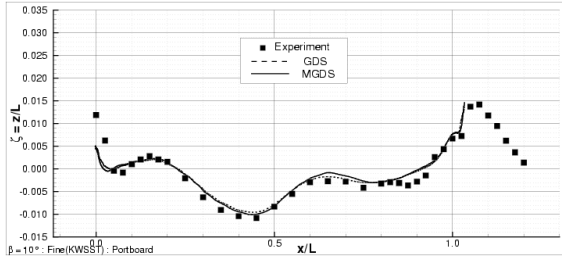
(a) 0 degree



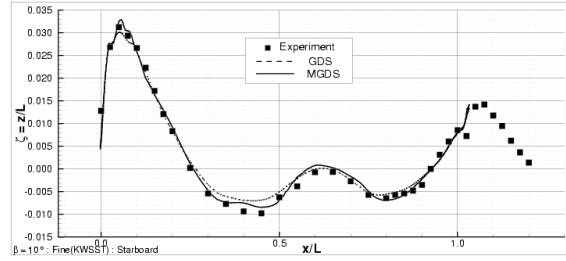
(b) 5 degrees - Port side



(c) 5 degrees - Starboard side



(d) 10 degrees - Port side

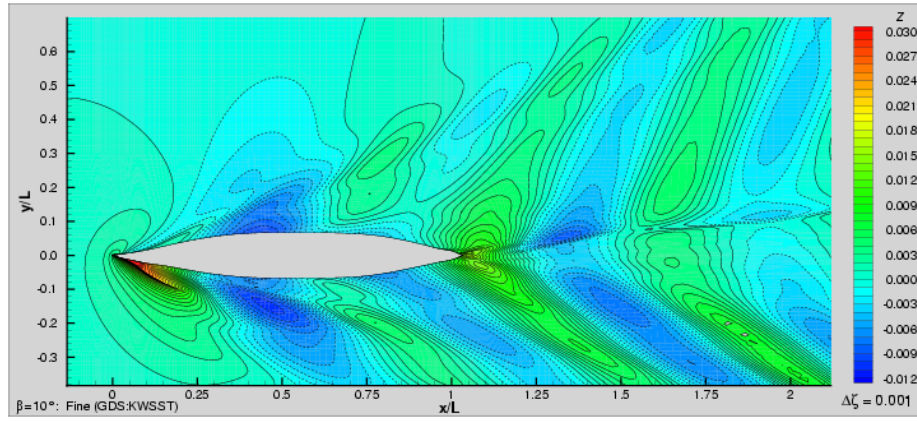


(e) 10 degrees - Starboard side

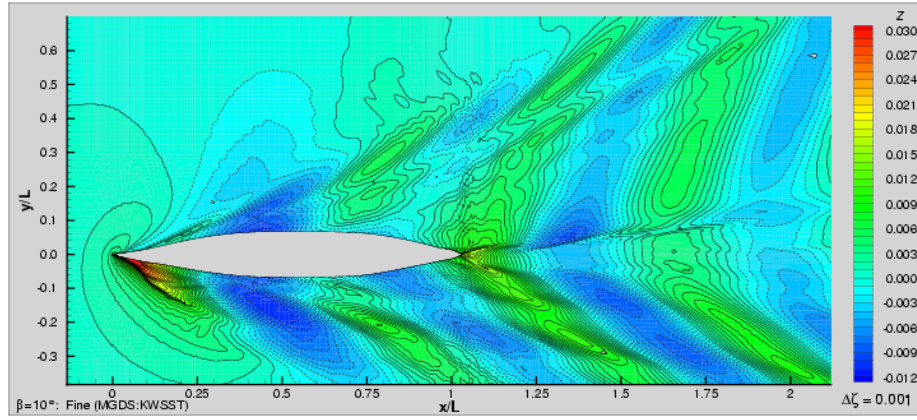
Figure 5: Free-surface elevation on the waterline. Comparison of GDS and MGDS discretisation schemes for different drift angles

3.4.2 Wave profiles cuts

Figs. 10 and 11 show comparisons between results obtained with GDS and MGS schemes to the experiments performed by IIHR (Longo & Stern, 2002) for drift angles of 5 and 10 degrees. Generally speaking, the agreement between computations and experiments is remarkable but one can notice the strong improvement brought by the MGDS scheme, because of its compressive property. This is particularly true for the transversal cut $X = 0.20$ in front of the hull where MGDS is the only scheme to detect a trend towards wave-breaking in agreement with the experimental findings. Transversal cuts in the wake of the ship ($X = 1.00, 1.20, 1.40$) reveal also a remarkable agreement with the experiments despite the fact that the grid density is likely not yet suited to the physical complexity of the stern flow.



(a) GDS



(b) MGDS

Figure 6: Free-surface elevations for a drift angle of 10 degrees. Comparison of GDS and MGDS discretisation schemes

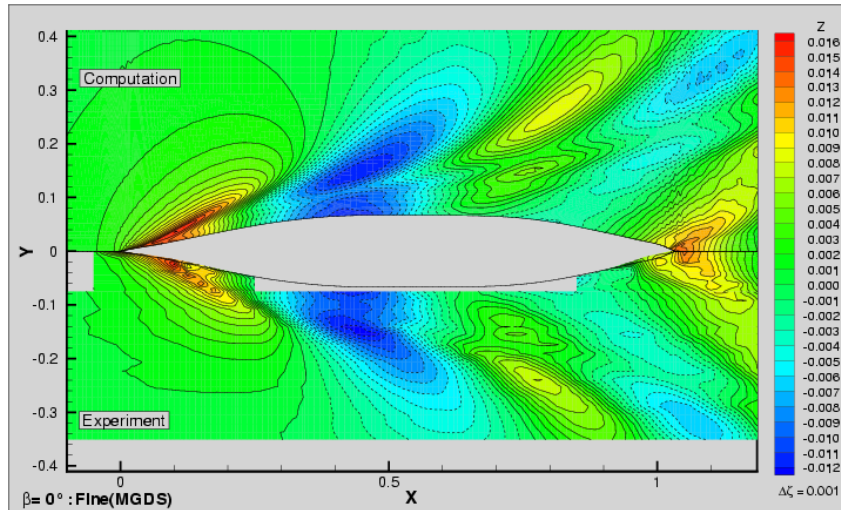
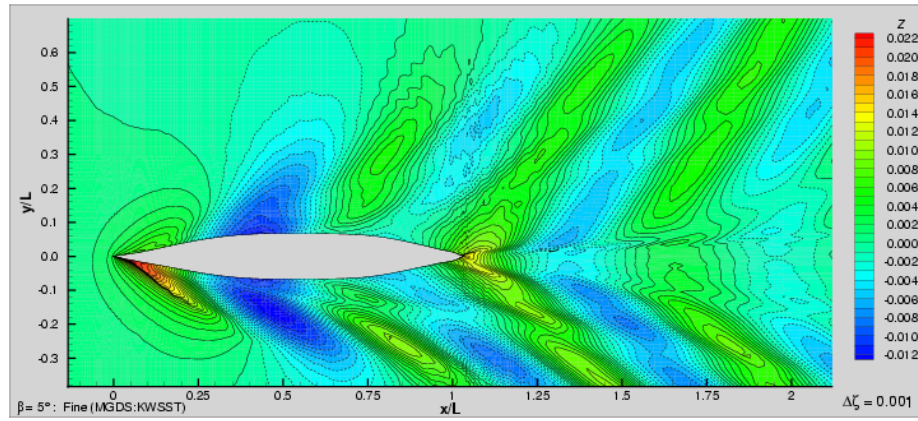
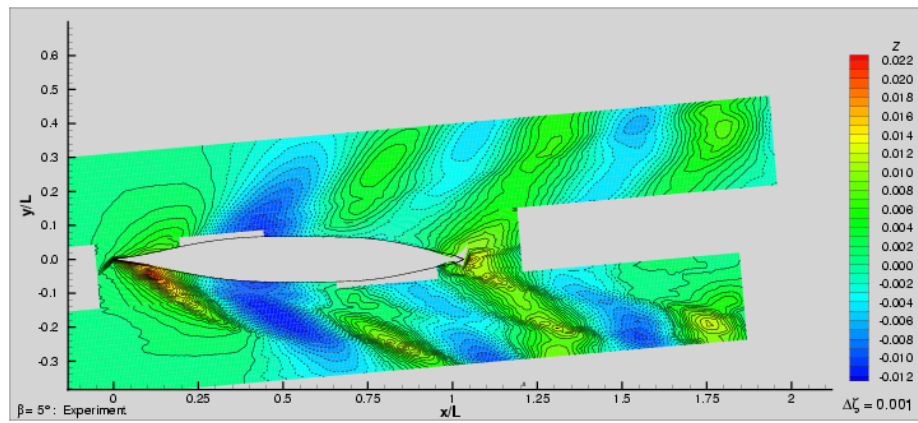


Figure 7: Free-surface elevations for a drift angle of 0 degree

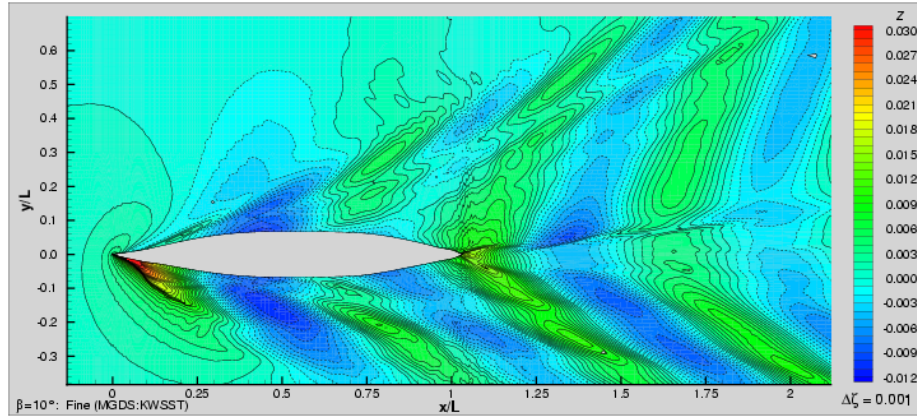


(a) Computations

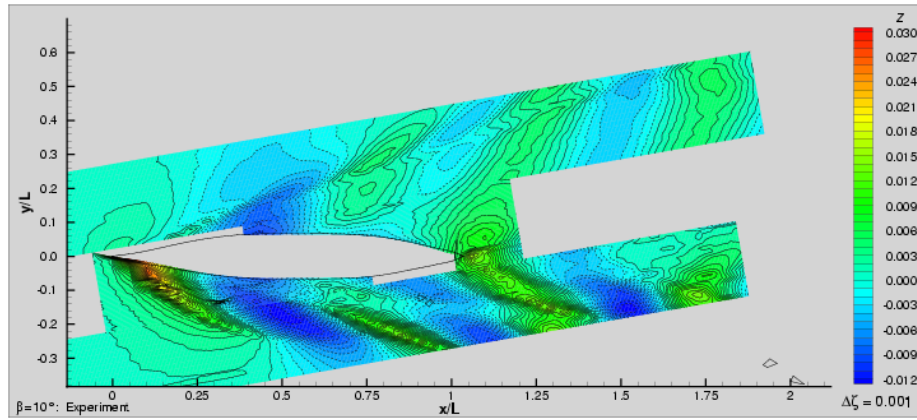


(b) Experiments

Figure 8: Free-surface elevations for a drift angle of 5 degrees

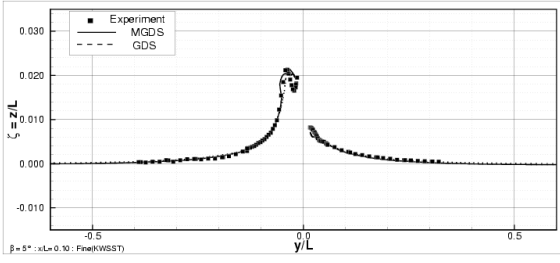


(a) Computations

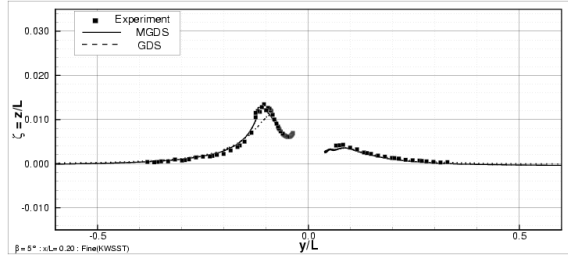


(b) Experiments

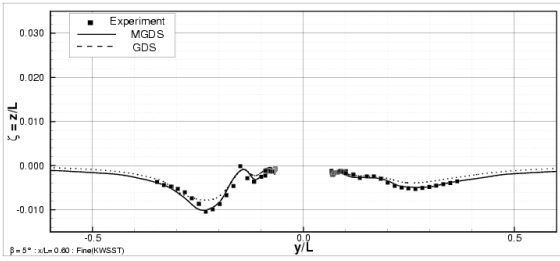
Figure 9: Free-surface elevations for a drift angle of 10 degrees



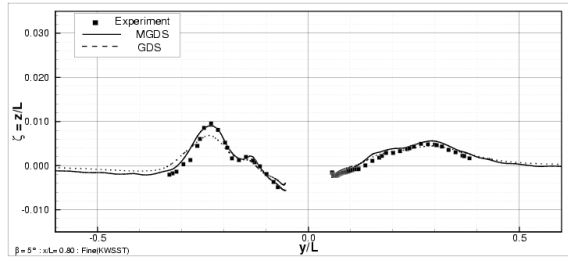
(a) $X/L = +0.10$



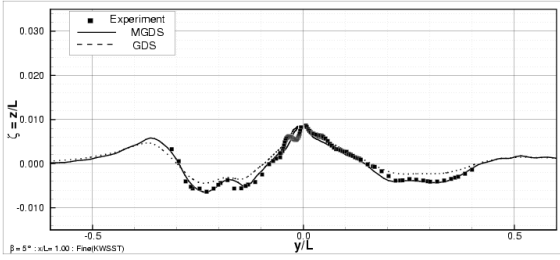
(b) $X/L = +0.20$



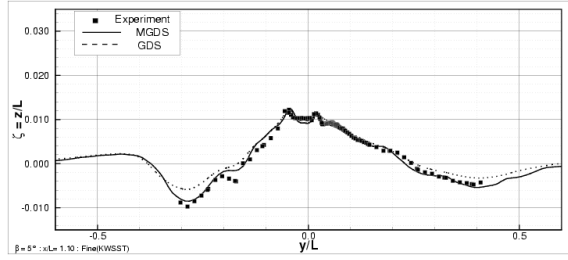
(c) $X/L = +0.60$



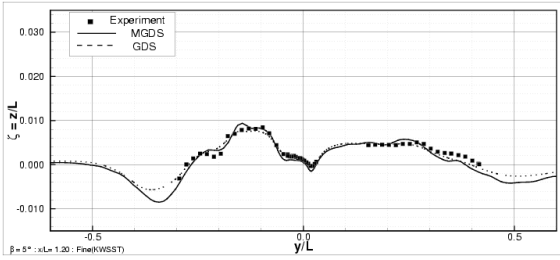
(d) $X/L = +0.80$



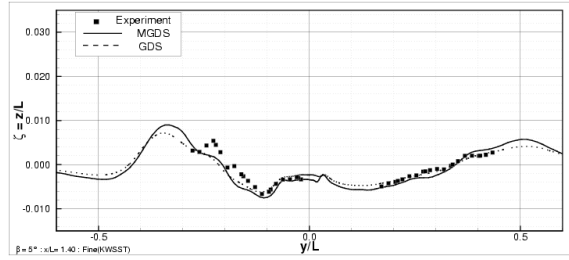
(e) $X/L = +1.00$



(f) $X/L = +1.10$

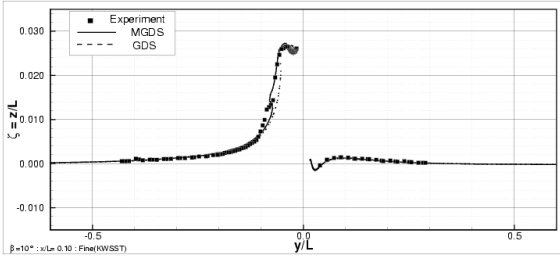


(g) $X/L = +1.20$

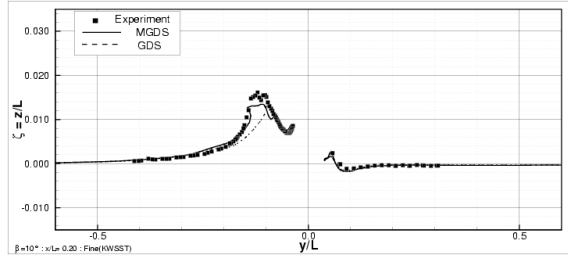


(h) $X/L = +1.40$

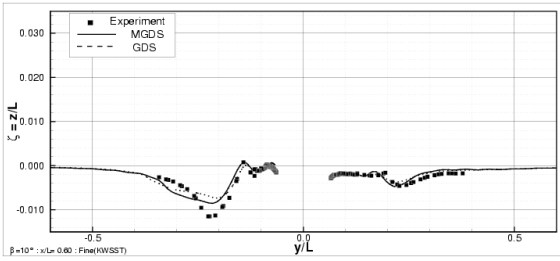
Figure 10: Wave profiles at $X=\text{cst.}$ for a drift angle of 5 degrees - Comparison between GDS, MGDS discretisation schemes and experiments



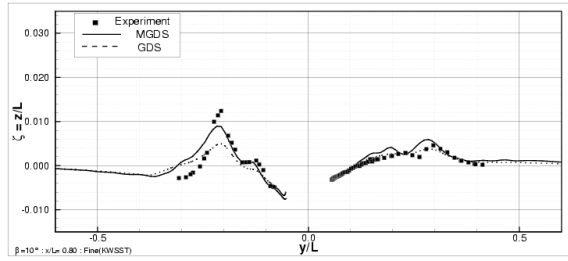
(a) $X/L = +0.10$



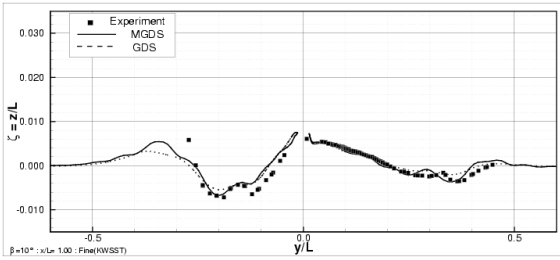
(b) $X/L = +0.20$



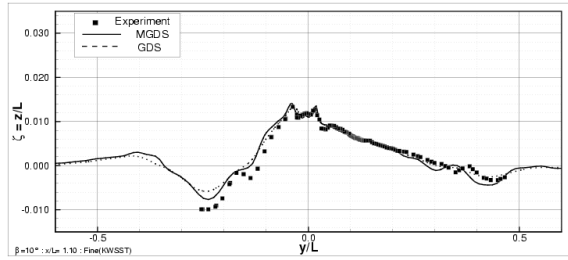
(c) $X/L = +0.60$



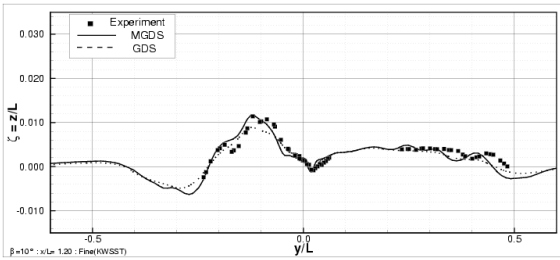
(d) $X/L = +0.80$



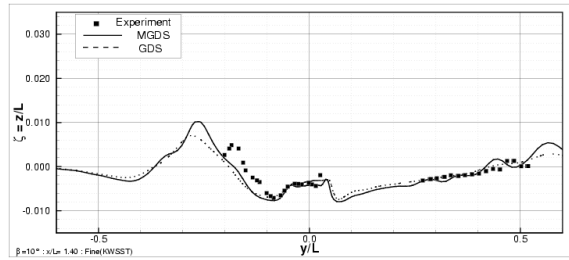
(e) $X/L = +1.00$



(f) $X/L = +1.10$



(g) $X/L = +1.20$



(h) $X/L = +1.40$

Figure 11: Wave profiles at $X=\text{cst.}$ for a drift angle of 10 degrees - Comparison between GDS, MGDS discretisation schemes and experiments

4 Conclusion

This paper has described a modern free-surface capturing strategy implemented in an unstructured finite-volume viscous flow solver. Special attention has been paid to the influence of the discretisation schemes of the transport equation for concentration and it was found that, despite the relatively fine grids used in this study, the role played by the compressive property is still fundamental to get a reliable simulation of the free-surface. A detailed comparison of computations versus experiments performed at IIHR showed that a quite satisfactory agreement has been reached, which makes this approach robust and reliable. Current efforts are devoted to the implementation of automatic grid refinement/coarsening strategies in order to put the grid points in accordance with a posteriori error estimation or explicit error indicators. Such a numerical strategy which couples a free-surface capturing approach to an automatic adaptive mesh refinement and coarsening methodology would be ideal since it would maintain dynamically (without human interference) a prescribed density of grid points around the steady or unsteady interface between air and water, without using too much grid points far from this interface.

5 Acknowledgments

The authors gratefully acknowledge the scientific committee of CINES (project dmn2050) and IDRIS (project 1308) for the attribution of CPU time.

References

- DEMIRDŽIĆ, I.; MUZAFERIJA, S. (1995), *Numerical method for coupled fluid flow, heat transfert and stress analysis using unstructured moving meshes with cells of arbitrary topology*, Comput. Meth. Appl. Mech. Eng., Vol. 125, pp.235–255
- DENG, G.B.; DUVIGNEAU, R.; QUEUTEY, P.; VISONNEAU, M. (2004), *Assessment of turbulence model for ship flow at full scale*, Comp. Mech., WCCM VI, Beijing
- DENG, G.B.; VISONNEAU, M. (1999), *Comparison of explicit algebraic stress models and second-order turbulence closures for steady flows around ships*, 7th Int. Conf. on Numerical Ship Hydrodynamics
- DUVIGNEAU, R.; VISONNEAU, M. (2003), *On the role played by turbulence closures in hull shape optimization at model and full scale*, J. Mar. Sci. Technol., Vol. 8, pp.11–25
- FERZIGER, J.H.; PERIĆ, M. (1996), *Computational methods for fluid dynamics*, Springer
- GASKELL, P.H.; LAU, A.K.C. (1988). *Curvature compensated convective transport: SMART, a new boundedness preserving transport algorithm*, Int. J. Numerical Methods in Fluids, Vol. 8, pp.617–641
- HIRT, C.W.; NICHOLS, B.D. (1981), *Volume of fluid (vof) method for the dynamics of free boundaries*, J. Computational Physics, Vol. 39, pp. 449–468
- JASAK, H. (1996), *Error analysis and estimation for the finite volume method with applications to fluid flows*, Ph.D. thesis, University of London.
- JASAK, H.; WELLER, H.G. (1995), *Interface tracking capabilities of the inter-gamma differencing scheme*, Internal Report, Mech. Eng. Dept., Imperial College of Science, London
- JASAK, H.; WELLER, H.G.; GOSMAN, A.D. (1999), *High resolution nvd differencing scheme for arbitrarily unstructured meshes*, Int. J. Numerical Methods in Fluids, Vol. 31, pp. 431–449

- LEONARD, B.P. (1988), *Simple high-accuracy resolution program for convective modelling of discontinuities*, Int. J. Numerical Methods in Fluids, Vol. 8, pp. 1291–1318
- LONGO, J.; STERN, F. (2002), *Effects of drift angle on model ship flow*, Exp. in Fluids, Vol. 32, pp. 558–569
- MUZAFERIJA, S.; PERIC, M. (1998), *Computation of free surface flows using interfacetracking and interface-capturing methods*, vol. chap. 3, pp.59-100, nonlinear water wave interaction ed., Comp. Mech. Publ., WIT Press
- PRŽULJ, V.; BASARA, B. (2001), *Bounded convection schemes for unstructured grids*, 15th AIAA Computational fluid dynamics conference, AIAA paper 2001-2593, Anaheim
- RHIE, C.M.; CHOW, W.L. (1983), *A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation*, AIAA Journal, Vol. 17, pp.1525–1532

State of the art in climbing and walking robots

**Manuel Armada, Pablo González de Santos, Manuel Prieto, Elena García, Teodor Akinfiev
Roemi Fernández, Hector Montes, Samir Nabulsi, Luis Pedraza, Roberto Ponticelli, Javier
Sarriá, Joaquín Estremera,** Instituto de Automática Industrial (CSIC), Spain, armada@iai.csic.es

Abstract

Modern robotic systems are increasingly powerful in terms of sensor fusion and mobility. The progress allows advanced robots to cope increasingly better also with complex environments which are frequently found in the maritime industries. An overview of the development of mobile robots (climbing and walking) is given with examples taken from the maritime industries applications coming mostly from the experience of IAI-CSIC in different projects. The overview article discusses past, present and prospects of walking and climbing robots intended to give an introduction to the exciting field of robotics to interested non-specialists.

1. Introduction

The Automatic Control Department of the Industrial Automation Institute (IAI-CSIC) has been carrying out research and development projects in the field of robotic systems for more than twenty five years. Since late seventies this activity began with the realisation of industrial robots, what provided the research team with a wide experience and reputation in robot kinematics, dynamics, mechanical design, and control systems. After some successful developments in that field, the department focused its interest in the area of robots for hostile/hazardous environments. In these kind of environments it is necessary to carry out a variety of tasks (inspection, manipulation, welding, grinding, etc.), what implies human operators are exposed to hard working conditions. Also there are a great number of potential applications that cannot be performed directly by human operators because of difficulties in reaching working positions in a proper and safe way. This situation yields, in a natural way, to the utilisation of remotely controlled devices, where tele-robots can be considered as the most advanced and promising solutions. Doing so, a number of advantages will come: improved working conditions, improved safety, automation of repetitive tasks, and opening the possibility of providing innovative solutions to emerging applications.

However, although many applications can be solved by means of an appropriate tele-manipulator, equipped with the right tools and with the concurrence of the human operator skills, many others cannot be solved in this way due to working position difficult access. The problem of accessing to more or less remote job sites presents major difficulties and prevents automation. There are, reported in the literature, interesting solutions to this situation, for example very long reach manipulators. Other, not less interesting approach, is to provide a transport mean for the tele-manipulator. Such a transport mean includes wheeled or tracked vehicles, and more recently, legged-machines (climbing or walking).

Nowadays shipbuilding industry is being forced to adapt its production to new technical specifications, shorter delivery time and new safety regulations, so that the ships have to be built faster, more economically and under better environmental condition for operators. New robotic systems are improving these features. Especially, robot manipulators are helping to enhance the quality of welding, decrease arc time, and avoid operators be exposed to fume concentration. However, current robotic systems cannot accomplish some industrial applications, especially those related with mobility in complex environments. These scenarios appear in some stages of the ship construction such as are the operations in the dry dock, and also in the ship repairing yards in what respects ship cleaning and inspection. In this paper some solutions are presented dealing with specially tailored climbing and walking robots for the maritime industries. These robotic systems have been mostly developed in the framework of European funded projects.

2. Climbing and walking robots for shipbuilding

In the field of shipbuilding there exists three main stages in the ship erection process:

- Block's construction in the workshop.
- Transportation of blocks to the dry dock or to the slip-way using cranes and especial vehicles.
- Connection of consecutive blocks in the dry dock or slip-way.

The first activity consists of the construction and assembly of huge ship blocks. This work is performed in highly automated workshops with a relatively good productivity, which is being increased by current research in this area. After the transportation of the blocks, which is performed in the second stage, the third involves joining two consecutive blocks by welding together all the longitudinal reinforcements and all the vertical bulkheads.

For environmental safety, most ships, especially tankers and bulk carriers are built with a double bottom and double hull so the cargo will not spill out if the hull is breached. This double structure forms cells all over the ship's hull. A typical tanker bottom cell can be up to 10 meters long, 4 meters wide and 3 meters high. To allow workers inside, cells have a manhole measuring about 0.8 x 0.6 meters.

There are two important welding problems in ship erection: butt-welding in position along the near-flat external hull surface, and butt/fillet welding for joining double hull cells (Figure 1). In the last years IAI-CSIC has been involved in several projects dealing with welding automation in shipbuilding. Four main results are briefly reported in this work: one six-legged and one four-legged climbing robots for butt-welding of ship hull skin, one four-legged walking robot for welding tasks inside ship hull double bottom, and one robotic system for welding inside the double hull vertical cells. All robots has been equipped with industrial welding units and special sensors for seam tracking.

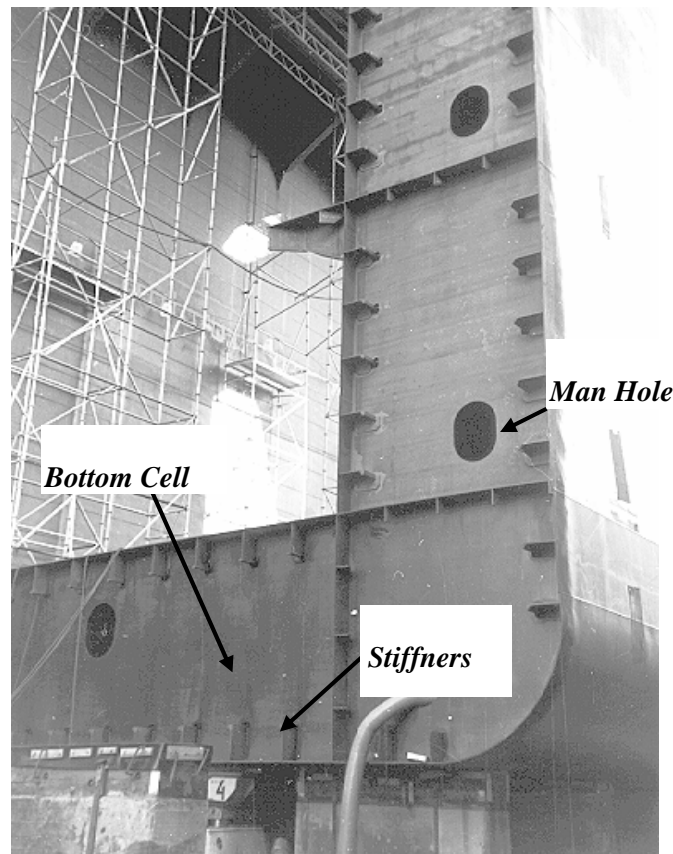


Figure 1: Dry dock ship block erection.

2.1 REST 1 and REST 2 climbing robots

The REST 1 climbing robot has six reptile-type legs with three degrees of freedom each one, actuated by dc motors through appropriate gearing. The leg kinematics is of *scara* type, with two rotational articulations and a prismatic one that holds at its end the foot. Feet at the end of legs are provided with special grasping devices based on electromagnets, securing the robot to ferromagnetic-material walls with intrinsic safety. Some degree of compliance has been provided to the feet, by means of an extra passive degree of freedom, so that the robot can adapt itself to a certain extent of surface unevenness. The climbing robot carries on board his control system that consists on an industrial PC that serves as a master for a bunch of slave processors that controls in real time the 18-degrees of freedom. One of the main features is the combination of 6 low-cost/high-performance digital control and 6 power electronic cards (one per leg, each one providing control for 3 joints), specifically developed for this project, and that are the responsible of the just mentioned control of each one of the 18 degrees of freedom of the robot.

The main specifications of the REST 1 climbing robot are:

- Leg number: 6
- Degrees of freedom: 18
- Body frame length: 1100 mm
- Body frame width: 600 mm.
- Robot weight: 220 Kg.
- Robot payload: up to 100 Kg.

Figure 2 left illustrates the experimental testing of REST 1 six-legged climbing robot. Different gaits and control algorithms has been implemented and evaluated.

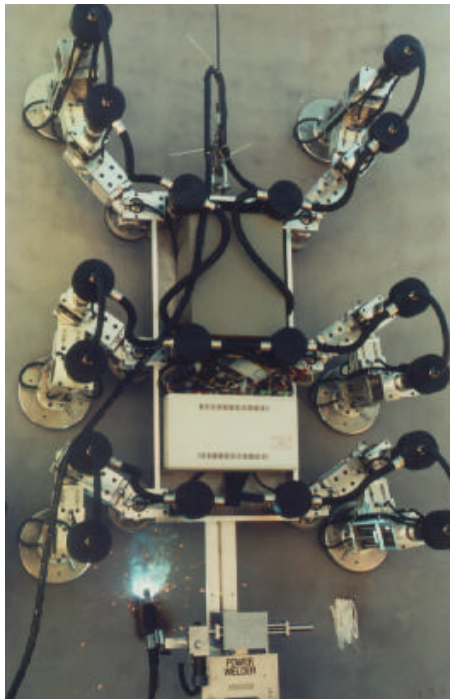


Figure 2: REST 1 climbing robot welding ship hull.

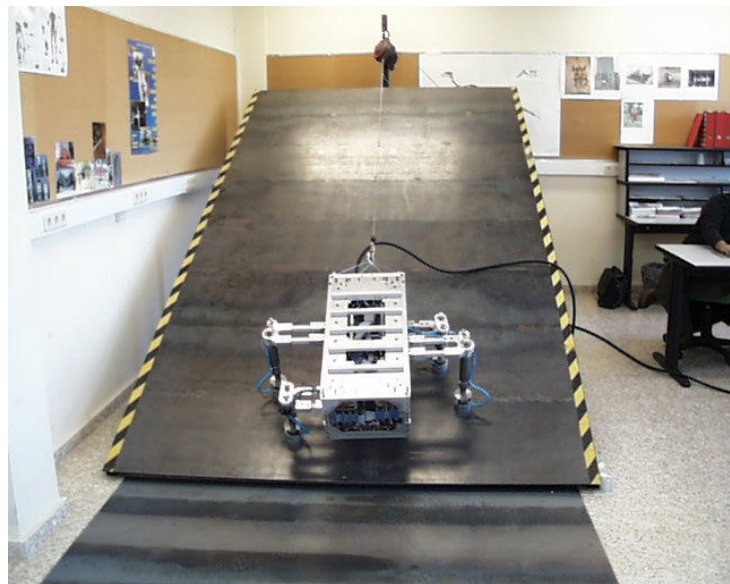


Figure 3: REST 2 climbing robot during testing

Following REST 1, a second prototype of climbing robot that moves continuously along the wall, named REST 2 has been constructed. It uses a variation of the well known wave gait in order to obtain fast continuous movement on softly undulated terrain together with foothold selection to handle obstacles and irregularities during climbing. The robot uses electromagnets to attach itself to ferromagnetic walls and has four legs (12 degrees of freedom) that resemble properties from sliding frames and true legged climbers. The novel leg design and geometrical configuration allows for fully overlapping workspaces. As a result of this novel leg and robot design, it was possible to achieve a much better payload to weight ratio, increased velocity, better inertial properties and reduced energy consumption. Our approach to reliability was to simplify the robot structurally and to support modularity of parts and connections.

REST 2 has been designed to carry on a light manipulator for butt welding. The robot weights less than 50 Kgs.

2.2 Rower 1 and Rower 2

The second development was intended for solving the problem of welding automation inside the double-bottom cells. The complexity of the working scenario and the system mobility requirements made a walking machine the best choice for this application. Different machine configurations were investigated in light of the end users' strict requirements, such as total weight, dimensions, payload, and assembly/disassembly capabilities. Finally, a four-legged machine able to walk by grasping cell stiffeners was envisaged as the best choice for our application. The ROWER 1 solution was required to be modular and the presence of manhole was very important for its design, because the volume of each subsystem should be constrained to the manhole's dimensions. Another constraint was that the weight of each subsystem (manipulator, welding system, mobile platform, vision system, etc.) should be less than 50 kg due to labour regulations on systems carried by operators. This is the reason why the mobile platform was divided into different parts and assembled in the cell.

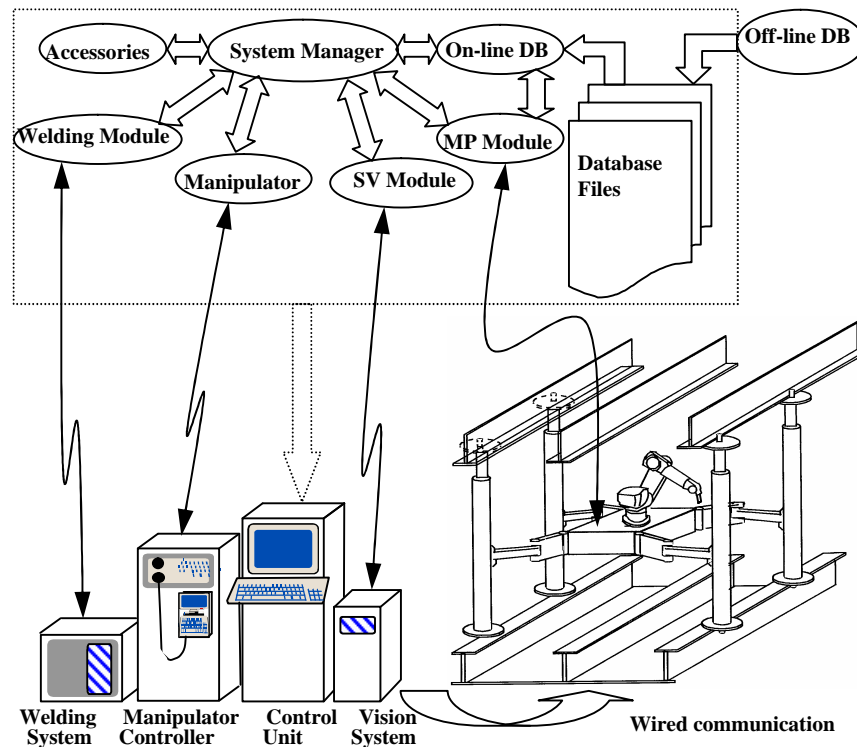


Figure 4: ROWER 1 system configuration.

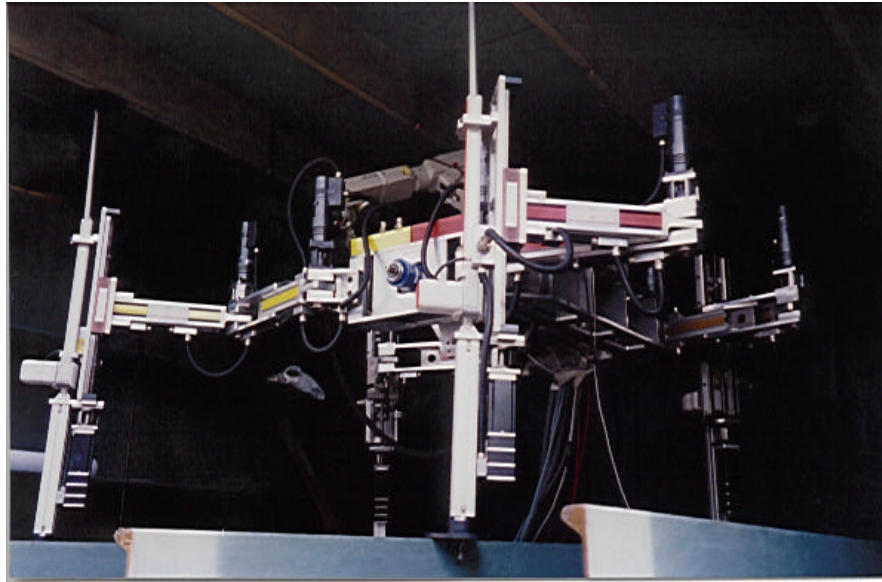


Figure 5: ROWER 1 system.

The full ROWER 1 system comprises the four-legged mobile platform (walking robot), which provides 12 degrees of freedom for displacement plus 4 degrees of freedom for “grasping” up/down the stiffeners, along with an industrial welding manipulator, stereo vision unit, and welding equipment. The ROWER 1 system is controlled remotely (around 30 m far from the robot) by means of a control unit located outside the double-bottom cell. Very complex software was prepared for this system. Figure 4 illustrates ROWER 1 organisation, and Figure 5 shows ROWER 1 mobile platform inside double-bottom cell at IAI-CSIC facilities.

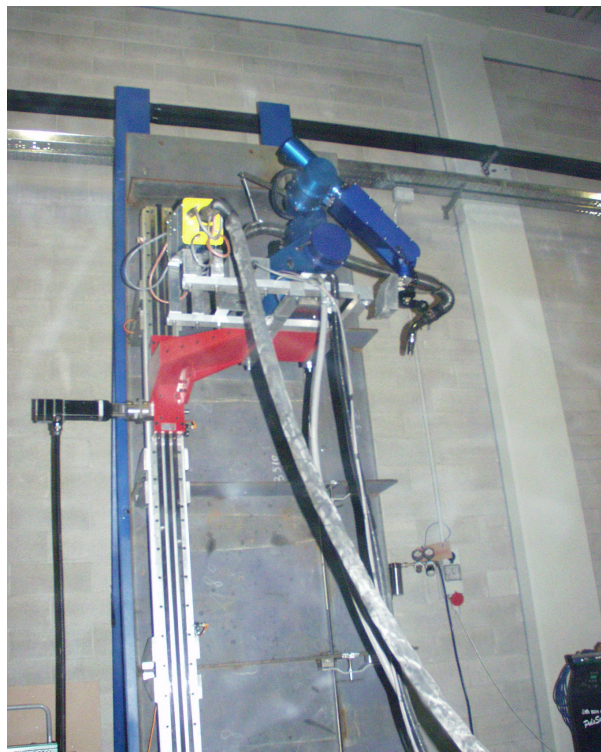


Figure 6: ROWER 2 system.

Further developments yielded to ROWER 2 (Figure 6), where a light manipulator is moved in the vertical direction for welding the double hull vertical cells.

3. Aurora underwater climbing robot

It is well known that all kind of ship's underwater hull become overgrown with sea adherence (weed, barnacles) very fast. This means raise of fuel consumption, and freeing atmosphere an extra amount of CO₂ (incrementing greenhouse effect) and of sulphur dioxide (acid rain), apart from deterioration of ability of ship's control. This situation becomes important even after six months of ship activity. For recovery of ship's required operational performance, it is necessary for Ship-Repairing and Conversion Industries to dry dock a ship and proceed to cleaning. This procedure is very time consuming and of high cost, but it is the only available solution nowadays for SRYs. On the other hand, this cleaning activity is the first to be done when a ship needs maintenance and/or some repairing, being the last the main activity of SRYs. So hull treatment is required and, at present time, is done manually in dry-dock using different adapted methods like grit blasting or water jet, and it has to be noticed that, in itself, it is a very contaminant operation (dust contains always painting particles), it is harmful for human operators health and it is a very uncomfortable job.

To provide a solution to these problems an EC funded project (G3RD-CT-000-00246) was organised: AURORA (Auxiliary Climbing Robot for Underwater Ship Hull Cleaning of Sea Adherence and Surveying). The project partnership brings together 7 partners with complementary roles: the Industrial Automation Institute (IAI-CSIC) which is the Project co-ordinator, two ship-repairing yards, T. Kalogeridis&Co. Inc. and Unión Naval de Barcelona, Algosystems S.A., the Division of Robotics, Department of Mechanical Engineering, from Lund University, SAIND, manufacturer and vendor of equipment for shipyards, and Riga Technical University.

AURORA scenario consists in the underwater hull that after some time of ship operation is plenty of marine incrustations, where a new kind of underwater climbing robot equipped with special tools should perform cleaning and surveying tasks. That scenario presents large dimensions and exhibits some areas of very difficult reach-ability and poses some additional technical difficulties. As it has been conceived the underwater climbing robot control is a human-in-the-loop process. Human-Machine Interface (HMI) design takes this into consideration whether using direct control or supervised control. This interface includes a control command set used to select the machine trajectory and a graphic representation used to get information about the robot and its environment. Figure 7 shows the concept of AURORA project and some view of the teleoperation station. The system has demonstrated excellent performance.

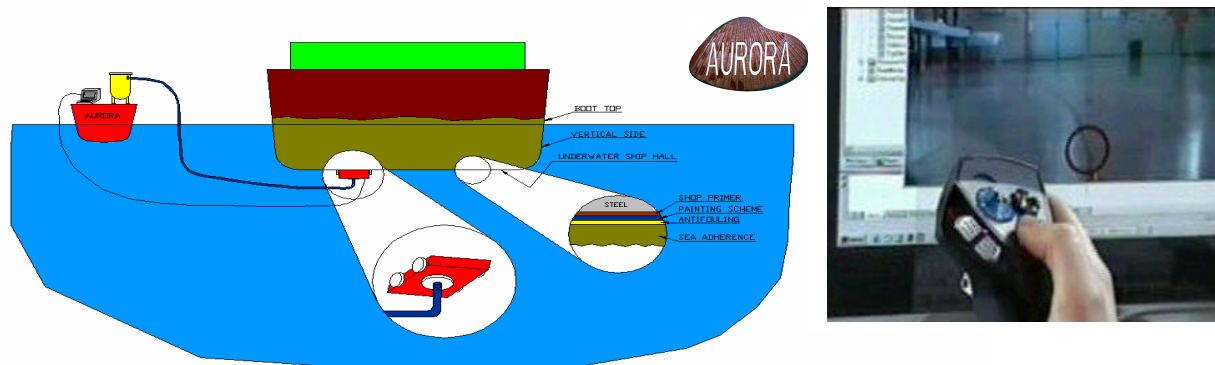


Fig. 7: AURORA concept and remote control

4. Conclusions

Some achievements in the field of climbing and walking robots in the last years have been presented. Most of the systems have being conceived to solve practical problems, but a lot of research is underlying and there are still many open questions.

Acknowledgements

The REST climbing robot has been developed entirely in the IAI under the project PACE PR 212 SACON funded partially by ESPRIT and by the CDTI-MINER of Spain. The authors want to acknowledge also the other two project partners AESA and SAIND for their co-operation. REST-2 was developed with CICYT funding under TAP 1999-0993 "SACON-2". ROWER 1 and ROWER 2 have been funded by BRITE/EURAM and GROWTH; other partners where TECNOMARE, FINCANTIERI, ENVC, CELSIUS, LUND University and AESA. IAI-CSIC is a member of the ROBMAR and CLAWAR Thematic Networks, where and important activity dealing with advanced concepts, technologies and applications of climbing and walking robots is being carried out. AURORA was funded also by the CE under GROWTH programme under contract G3RD-CT-000-00246. Author's acknowledgement is extended to all the partners and to the European Commission

References

- JONES, D. (2000), *Afloat maintenance, the control of marine fouling and the care of coating underwater*, in "Marine Coating", 32nd WEGEMENT Summer School on Marine Coatings, Edited by David Short, pp.205-217
- NECSULESCU, D. S.;KIM, B.; REYNAUD, H. (1993), *Work space control of a mobile robot using range and contact force sensing*, 1st IFAC International Workshop. Intelligent Autonomous Vehicles. University of Southampton, Hampshire, United Kingdom, pp.18-21
- SHIM, H.-S.; KIM, J.-H. (1994), *Robust Adaptative Control for Nonholonomic Wheeled Mobile Robot*, Proc. IEEE Int. Conf. On Industrial Technology, Guangzhou
- KANAYAMA, Y.; KIMURA, Y.; MIYAZAKI, F.; NOGUCHI, T. (1990), *A Stable Tracking Control Method for an Autonomous Mobile Robot*, Proc. IEEE Int. Conf. Robotics and Automat., pp.384-389
- AKINFIEV, T; ARMADA, M.; FERNÁNDEZ, R. (2001), *Vehicle Control Method*, Patent Application N° P200102427, Spain
- ARMADA, M. AND GONZALEZ DE SANTOS, P. (1997), *Climbing, walking and intervention robots*, in Industrial Robot, vol. 24, n. 2, pp. 158-163.
- P. GONZÁLEZ DE SANTOS, M. ARMADA, M.A. JIMÉNEZ (1997), *An Industrial walking machine for naval construction*, in IEEE International Conference on Robotics and Automation, Albuquerque (USA).

Global Optimization Algorithms in Naval Hydrodynamics

Antonio Pinto, Daniele Peri, Emilio F. Campana, INSEAN, A.Pinto@insean.it

Abstract

Real-life optimization problems are much more complicated than the analytical test functions usually adopted for estimating the performances of new optimization algorithms. This simple statement may sound like an obvious objection to most of ship designers as well as an outrageous criticism to the mathematical community. Anyway it is evident that for a numerical optimization procedure to be useful in the design spiral it is fundamental to understand the peculiar features of the design problem and choose the more suitable numerical technique among a bunch of methodologies. Unfortunately, although the number of paper related to numerical ship design is rapidly growing, there is still the need of testing well assessed algorithms in specific ship design applications, especially if talk about Global Optimization techniques. In this paper a certain number of Global Optimization algorithms are presented and discussed, showing results on a real ship design application. The focus is on the development of deterministic methods, which are preferred for their good theoretical properties and for the reduced number of objective function evaluations required, a fundamental property any method should have to deal with numerical optimization of ships.

1 Introduction

In the last decade, a growing attention has been paid on numerical optimization in the naval field and papers have been presented related to some applications, ranging from building cost of a ship to their hydrodynamic and structural characteristics. All these problems are highly constrained and an optimum solution is difficult to be found due to the nature of the problem itself. To attacks these problems, designers would become interested in Global Optimization (GO) in the next few years. Indeed, a simple local optimizer could be inefficient: in fact, local methods usually applied tend to be trapped into the basin of attraction of local minima without giving any estimate of the absolute minimum of the objective function. In such a situation, there cannot be any guess about the extra improvement one could have obtained if a different starting point would have been adopted. On the contrary, GO algorithms have the ambition to find the global optimum of the objective function, so that no further improvement can be obtained regardless to the starting point.

2 Test case

In order to evaluate the capability of different GO algorithms to find the global minimum of a ship design problem, a test case has been defined. The goal is to minimize the vertical motion of the center of gravity (heave motion) of a ship advancing at the speed of 16 knots ($Fr = 0.198$) in head seas. The minimum is searched for non-dimensional frequencies higher than 0.4 since for lower frequency values a response equal to 1 is the limit value regardless the hull shape. The Parent Hull Form (PHF) is the S175 containership (figure 1): it has been adopted by the ITTC seakeeping committee in a benchmark procedure and it may be a good test for this problem. To give the problem the characteristics of a real design case some geometrical constraints have to be enforced too. The choice in this case has been the following:

- Displacement: between 24000 and 24600 t
- Beam: between 25 and 26 m

About the numerical solver used to guide the optimization algorithm, being the focus of this

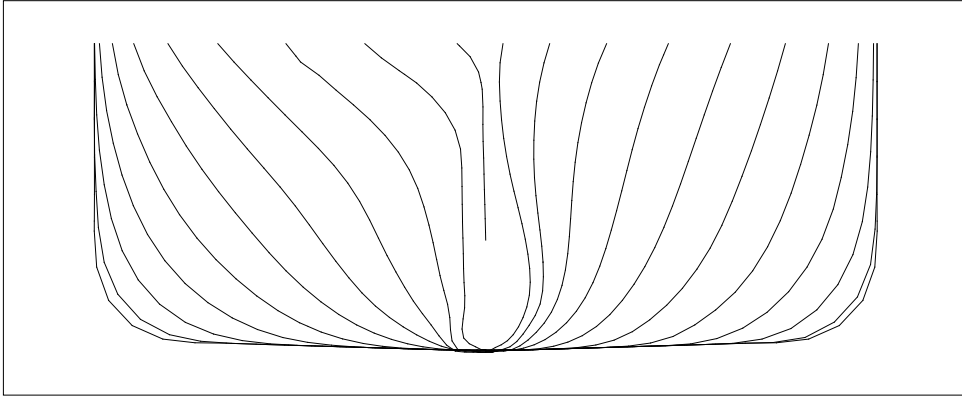


Figure 1: Section lines of the S175 hull.

test the analysis of the capabilities of GO methods, a simple and fast strip-theory solver for the seakeeping problem has been adopted. Hull shape parameterization has been performed according to *Peri et al. (2001)* and a Béziér patch has been superimposed to the original hull shape. Only six control points of the patch have been used as design variables. The advantages of this parameterization technique is its great flexibility in terms of construction of new shapes and ease of implementation, and the accuracy in the reproducing the exact initial hull shape. The patch acts in the y (transversal) direction only. As a consequence, the keel line is left unvaried during the optimization procedure.

3 Optimization algorithms

In this paper five different algorithms are tested: four GO optimization techniques and one local search algorithm used as representative of the behavior of this class of methods. The details of these methods are given in the following.

3.1 Conjugate Gradient Method

This method is largely adopted in a simple local approach: a sequence of line searches along descent directions are performed. At the first step, the descent direction is determined by the opposite of the gradient of the objective function $-\nabla f$, while from the second step on a conjugate direction is applied (the method is largely discussed in literature, see as an example, *Bonnans et al. (2003)*).

$$d_i^k = -g_i^k + d_i^{k-1} \frac{|g^k|^2}{|g^{k-1}|^2} \quad (1)$$

Due to its local nature, the algorithm proceeds straight to the first local minimum. Unfortunately, since the global optimum rarely is located in the neighborhood of the starting point, the search is usually very limited and no suggestions are given for any further possible improvement. Moreover, if the objective function is noisy, this method may be easily trapped by false minima created by the numerical noise.

3.2 Simplex Method

This method, proposed by *Nelder and Mead (1965)*, is usually treated as a local method. However, since the wideness of the investigation region for the generation of new points in the algorithm is user-defined, the method demonstrate to be able to overcome the problems connected with high-frequency noise of the objective function. A weak point of this method is that the

final convergence is slow: this means that, once the basin of attraction of the minimizer has been identified a pure local search method may be adopted to find the minimum

3.3 Parameter Space Investigation (PSI)

PSI method has been proposed by *Statnikov and Matusov (1995)*. It provides the Pareto Optimal Set (POS) of a multi-criteria optimization problem sampling the feasible region by means of an uniformly distributed sequence of points. The constraints are treated explicitly: this means that there are no difficulties related to their treatment. With respect to the original implementation, described in *Peri and Campana (2003b)*, some additional features for a more efficient determination of the minimum, and also to reduce the computational effort, have been proposed by *Peri and Campana (2004)*. Although the method is designed to solve multi-objective optimization problem, it is also applicable for single-objective problem. The method has been also validated through experimental campaigns (*Peri and Dattola (2003)*).

3.4 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a pure stochastic method introduced for the first time by *Kennedy and Eberhart* in 1995. The PSO is an evolutionary technique even though it differs significantly from classical genetic algorithms. Being a more recent and less diffused technique we present in the following the general framework of the method, while extensions and discussions of specific issues may be found in the cited references. A population of trial solutions (ideally, particles forming a swarm) is used to sample the design space, each particle having an adaptable speed according to both individual and collective rules. In PSO there are no DNA inspired operators applied in the swarm. Instead, original PSO formulation defines each particle as a potential solution of a problem in a N -dimensional space, where the i^{th} particle of the swarm can be represented by a N -dimensional vector, $X_i = (x_{i1}, x_{i2}, \dots, x_{iN})^T$. The velocity of this particle is given by the vector $V_i = (v_{i1}, v_{i2}, \dots, v_{iN})^T$. The best previously visited position of the i^{th} particle is denoted as $P_i = (p_{i1}, p_{i2}, \dots, p_{iN})^T$. Defining b as the index of the best visited location by the whole swarm, and letting the superscript denote the iteration number, the swarm is moving according to the following two equations (*Shi and Eberhart (1998)*, *Shi and Eberhart (1999)*):

$$v_{id}^{n+1} = \chi [\omega v_{id}^n + c_1 r_1^n (p_{id}^n - x_{id}^n) + c_2 r_2^n (p_{bd}^n - x_{id}^n)] \quad (2)$$

$$x_{id}^{n+1} = x_{id}^n + v_{id}^{n+1} \quad (3)$$

where $d = 1, 2, \dots, N$; $i = 1, 2, \dots, n_{sw}$, and n_{sw} is the size of the swarm; ω is called *inertia weight*; c_1, c_2 are two positive constant, called *cognitive* and *social* parameter respectively; χ is a *constriction function*, which is used to limit the velocity; r_1, r_2 are random numbers, uniformly distributed in $[0,1]$; and $n=1,2,\dots$, determines the iteration number.

The role of the *inertia weight* ω , in eq. 2, is considered critical for the PSO's convergence behavior. The inertia weight is employed to control the impact of the previous history of velocities on the current one. Accordingly, the parameter ω regulates the trade-off between the global (wide-ranging) and local (nearby) exploration abilities of the swarm. A large inertia weight facilitates global exploration (searching for new areas), while a small one tends to facilitate local exploration, i.e. fine-tuning of the current search area. A suitable value for the inertia weight ω usually provides balance between global and local exploration abilities and consequently results in a reduction of the number of iterations required to locate the optimum solution. Experimental results indicated that it is better to initially set the inertia to a large value, in order to promote global exploration of the search space, and gradually decrease it to get more refined solutions (*Shi and Eberhart (1998)*, *Shi and Eberhart (1999)*). An initial value for ω was set and the gradual decrease was calculated as follows:

$$\omega = \omega_0 \left(1 - \frac{n-1}{n_{max}} \right) \quad (4)$$

The parameters c_1 and c_2 , are not critical for the PSO's convergence. However, proper fine-tuning may result in faster convergence and alleviation of local minima.

Starting from this *standard* formulation, a new formulation has been here derived, by introducing the modifications described in the following.

3.4.1 Deterministic flavor

In this case, we decide to set $r_1 = r_2 = \epsilon$ ($0 < \epsilon \leq 1$), thus eliminating the random factor introduced by the two coefficients. In this way we transformed a pure stochastic method in a deterministic one. to have more control on the process and the possibility of comparing different runs.

3.4.2 Additional group velocity

As all the inner methods, PSO tends to implode, searching the optimal solution inside the assigned domain. As a consequence, it is quite appropriate to locate initially the swarm on the boundaries of the design space. But, due to the adopted parametrization, it is quite difficult to find out a direct correspondence between the value of the design parameters and the geometrical constraints and to fix the bound on the design parameters correspondingly. In the implementation, to avoid the exclusion of some interesting part of the feasible set the bounds must be set as large as possible, keeping in mind however that a too large domain means a excessive number of objective function evaluations. To this aim, a term has been added to equation 2 to account for a kind of group velocity. This term is computed tracking the speed of the geometrical center of the swarm. An upper limit to the group velocity is applied as well: this limit decreases with the iterations.

3.4.3 Initialization procedure

The initialization of the swarm is considered as an issue of crucial importance for the PSO's performances in terms of correct localization of the global minima and low number of function evaluations. Thus we decide to use three different distributions:

- One particle at the center of each hyper-rectangle of the N -dimensional hyper-cube (2^N particle)
- One particle per corner and one at the center of the N -dimensional hyper-cube ($2^N + 1$ particle) - 2^N factorial design
- A LP τ -net (24 particle)

Hyper-cube and 2^N factorial design have been applied to the first modified algorithm, while the LP τ -net has been applied to the second modified algorithm. A picture view of the 2-dimensional case is reported in figure 2.

3.4.4 Stopping criteria

A robust convergence criterion is important for any general-purpose optimizer. Most implementation of the PSO algorithm also apply some convergence criterion. A convergence criterion is necessary to avoid any additional function evaluations once an optimum solution is found. Since we miss any information about the global minima and we are also unable to certify the covering of the design variable space, unlike in the uniform covering methods, we have no information about the uninvestigated design space regions. Stationary point is not detectable as well, since

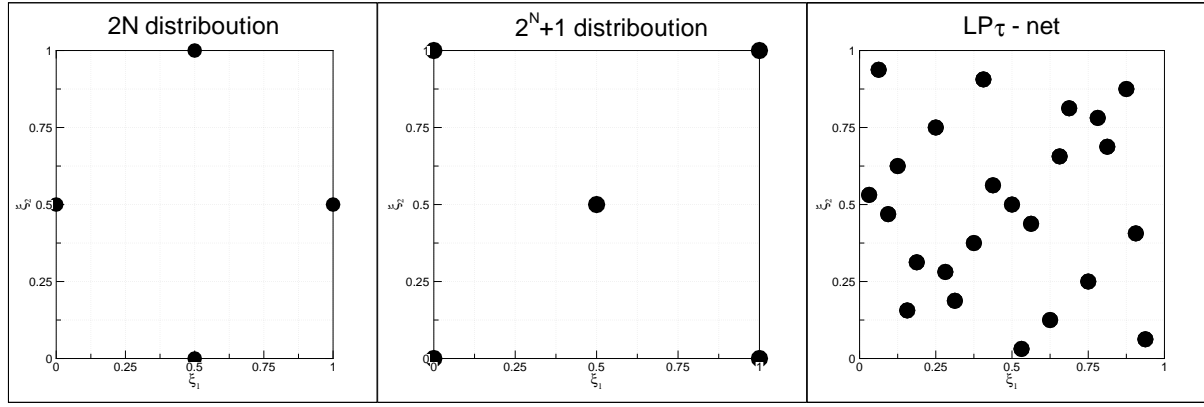


Figure 2: The three different initial configurations of the swarm particles for a 2-dimensional case.

local gradient of the objective function is not computed. As a consequence, the convergence criterion used here simply stops the research when a fixed number of function evaluations is exceeded. The diameter of the swarm could be also adopted as an indicator for stopping the process.

3.5 *Lipschitzian Optimization* method (Diagonal Rectangular Analysis for Global Optimization - DRAGO)

Certification of the global minimum is difficult to obtain and these methods try to maintain some theoretical properties by operating an uniform covering of the design space. Obviously, a trade-off between certification and CPU time is necessary. A wide variety of different methods aimed to uniformly covering the design variable space, usually addressed as covering methods, has been proposed (*Törn and Žilinskas (1989)*). Some of these methods require an estimation of the Lipschitz constant K (the maximum possible slope of the objective function in a given interval) in order to evaluate the infimum of the objective function inside a bounded region (see *Shubert (1972)*, *Jones et al. (1993)*). Successive further subdivisions of an initial hyper-rectangle, including the feasible region, are performed on the basis of the width of each interval and of the value of the objective function at the bounds. These methods differs from each other in the strategy of selection of the interval to be divided and in the strategy of estimation of the Lipschitz constant. Anyway, most of the methods requires at least 2^N objective function evaluation, since the vertex of the N -dimensional hyper-rectangle bounding the feasible solution set are needed. If CPU-time hungry codes are applied as analysis tools and the number of design variables N is high, the initialization phase may require a huge amount of time, and the solvability of the problem may become critical.

To this aim, the Shubert algorithm (*Shubert (1972)*) has been modified. Each hyper-rectangle is characterized by two points only, that is, the extreme corners (all variables set to the lower and the upper bound, respectively). The initialization is made by computing the objective function in these two vertex of the original hyper-rectangle, regardless the dimension of the problem. Then, considering each design variable separately, the infimum of the objective function over each hyper-rectangle is estimating by applying:

$$y_i = (f_i^u + f_i^l)/2. - K * (u_{ij} - l_{ij}) \quad (5)$$

where i denotes the interval index, j the design variable, u_i and l_i the upper corner of the i^{th} hyper-rectangle, f_l and f_u the objective function value at l_i and u_i , K the Lipschitz constant

and y_i the estimated infimum of the objective function inside the i^{th} hyper-rectangle. The hyper-rectangle to be divided, along a particular j^{th} direction, is detected. For each subdivision, two new hyper-rectangles are obtained, and only two new values of the objective function are necessary (figure 3).

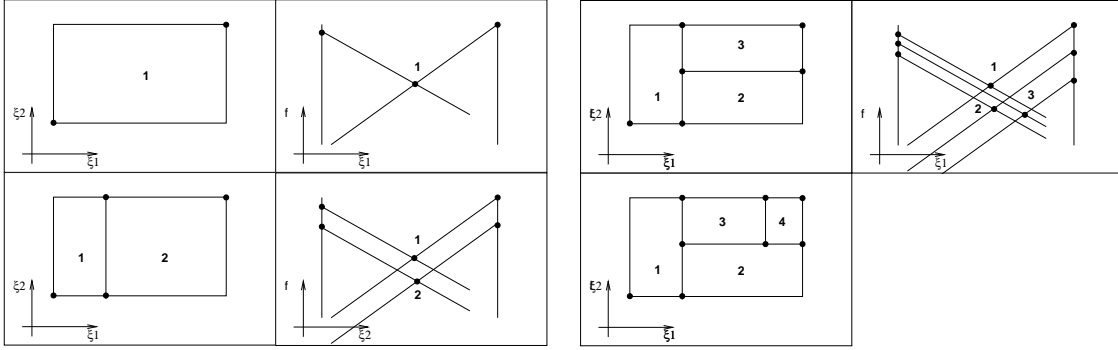


Figure 3: Course of the DRAGO algorithm.

The Lipschitz constant is estimated as the maximum value of the ratio of change of the objective function along each interval augmented by a certain factor. After a fixed number of iterations (subdivisions), the bounding interval is reduced, centering on the actual best solution. The process continues by subdividing all the hyper-rectangles belonging to these new limits.

4 Results

The methods described in the previous paragraphs have been tested and optimal configurations have been found, obtaining different optimal hulls. In figure 4 the Response Amplitude Operators (RAO) associated to the various hullforms are reported. Values of the objective function improvements and of the number of required evaluations are reported in table 1 for the sake of comparison, while the renderings of the hullforms are reported in figure 5. In figure 7, history of the design variables is reported as a function of the iteration step.

Table 1: Comparative results for the heave motions optimization.

Algorithm	Heave	Improvement	N_f
Original	1.320	-	-
Conjugate Gradient	1.190	9.84%	234
PSO ($n_{sw}=12$)	0.901	31.74%	300
Simplex	0.897	32.04%	1230
PSO ($n_{sw}=65$)	0.888	32.72%	1950
DRAGO	0.883	33.10%	1670
PSI	0.882	33.18%	1800
PSO ($n_{sw}=24$)	0.868	34.24%	1500

A strong gap between the local method and the performances of the global methods is evident. The local gradient-based method remains trapped in the neighborhood of the original configuration, and the improvement is very small when compared with the global methods.

The reason of the shortness of the path covered by the gradient method is explained in figure 6. Here some hyperplanes of the objective function around the minimum value are reported. On each hyper-plane, a couple of design variables are changing while the other four design variables

are kept at a constant value. Solid thick lines are the iso-levels of the objective function, while the dark regions indicate the feasible part of the hyper-plane. The extension of the hyper-plane is small in comparison with the dimension of the area to be investigated during the optimization process: here each design variable can assume values between $[-20:20]$, and the hyper-plane is centered on the optimal value and has an extension $[-1:1]$. Moreover, the feasible region is narrow, and it is also nearly parallel to the contour levels of the objective function. As a consequence, the local gradient is orthogonal to the feasible region and this causes a great difficulty for gradient-based methods in finding a descent direction compatible with the constraints. Finally, the objective function is globally smooth, but locally noisy, and also this feature causes difficulties in the correct estimation of the local gradient. This last feature is quite common for CFD-based optimization.

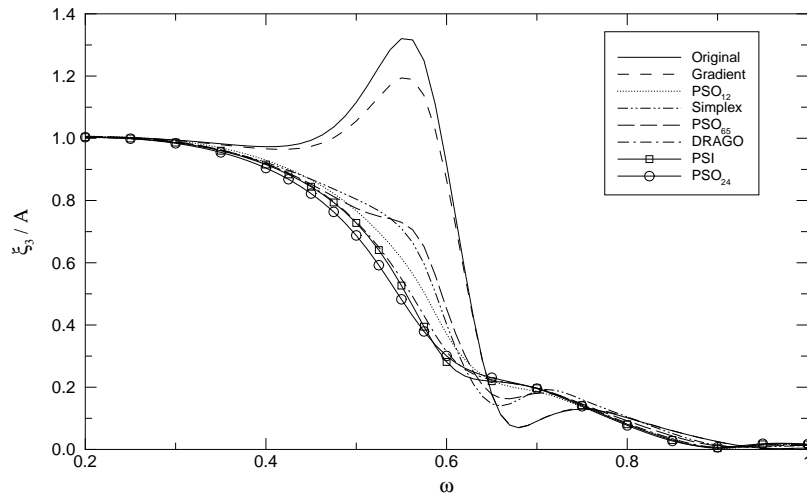


Figure 4: Comparison of the RAOs of the original and optimized hulls.

Although all these difficulties, an improvement of about 10% has been obtained by the gradient-based method. However, all the GO methods are able to improve of more than 30% the objective function. Among these methods, also the Simplex method is listed, since this descent strategy is not based on the gradient of the objective function: moreover, if the simplex dimension is big enough, the method performs a wide-range search in the design variable space. This does not mean that this method is a real GO method, but its features are closer to the GO than to the classical local search methods.

Both PSI and DRAGO attempt a uniform covering of the design space and the obtained results reflect this goal. Both the algorithms tend to localize the search during the optimization process: the obtained results denote a fast localization of the more promising region. Moreover, PSI operate an approximation of the objective function in order to localize promising regions: this way, there is a filtering effect on the objective function, and sharp attraction basins are very difficult to be detected.

Among the investigated algorithms, PSO seems to give the more interesting results and the investigation capability of the PSO seems to be improved by adding a group velocity to the swarm particles. Looking at figure 7, it is clear that the gradient-based method is trapped in the nearby of the initial solution. Simplex method moves gradually to the proper minimum, but the final value of the design parameters is closer to the starting point if compared to the other methods. PSO₁₂ and PSO₆₅ and DRAGO stay inside the original limits for the design

parameters, while PSI and PSO₂₄ are able to overcome this barrier.

The narrow shape of the feasible set create difficulties to the algorithms: narrow feasible regions are uneasy to detect. PSI could be partly unaffected by this situation since feasible points are placed all along the design variable space, but a high density of trial solution are necessary for a detailed description of the feasible set. Constraints evaluation requires a small time, but if the number of configurations to be verified is huge, the amount of time for the set-up could be comparable with the time necessary for the optimization as a whole. Regarding the search of the feasible solution set, inefficiency of the PSO may be addressed to the particle velocity: if it is too high, small feasible region are missed, while a too slow velocity may affect the rate of convergence.

The highest quality of the solution is payed by a higher number of evaluations of the objective function. PSO₂₄ requires about 1500 evaluations while conjugate gradient requires only 234 evaluations. On the other hand, improvement on the objective function pass from 9.5% to 34.2%. Anyway, PSO₁₂ is able to give an improvement of about 31.7% with a modest amount of objective function evaluations (300). For this reason, the computational effort with the PSO could be considered comparable with the one required by a local search. On the contrary, PSI and DRAGO tend to uniformly cover the feasible space: as a consequence, the number of function evaluation cannot be small. Anyway, DRAGO gives an estimation of the quality of the obtained minimum, allowing a certification of the global minimum.

5 Conclusions

Five different optimization algorithm have been tested on a real-life optimization problem for a containership and GO methods proved to be much more attractive than a simple local optimization. Improvements on the original hullshape have been obtained showing a factor of 3 between local and global optimization solutions. On the other hand, an increase on the required number of evaluations of the objective function is connected to the quality of the final solution, and a factor in between 1.3 and 6 for the number of required evaluations is obtained w.r.t. the local optimizer. Among the GO techniques, PSO seems to be able to close the gap, requiring the lowest number of objective function evaluations with a good optimal solution.

References

- BONNANS, J.F.; GILBERT, J.C.; LEMARÉCHAL, C.; SAGASIZÀBAL, C.A (2003), *Numerical Optimization - Theoretical and practical aspects*, Springer
- JONES, D.R; PERTTUNEN, C.D.; STUCKMAN, B.E. (1993), *Lipschitzian Optimization Without the Lipschitz Constant*, J. Optimization Theory and Application, Vol. 79, No. 1, October 1993, pp.157-181
- KENNEDY, J; EBERHART, R (1995), *Particle Swarm Optimization*, Proc. IEEE Int'l. Conf. on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, IV:1942-1948
- NELDER, JA; MEAD, R (1965), *A simplex method for function minimization* Computer J., Vol. 7, pp.308-313
- PERI, D; ROSSETTI, M; CAMPANA, E.F. (2001), *Design optimization of ship hulls via CFD techniques*, J. Ship Research, **45**, 2, pp.140-149
- PERI, D.; CAMPANA, E.F.; DI MASCIO, A. (2001), *Development of CFD-based design optimization architecture* 1st MIT Conf. on Fluid and Solid Mechanics, Cambridge
- PERI, D.; CAMPANA, E.F. (2003a), *High-Fidelity models in the multi-disciplinary optimization of a frigate ship* 2st MIT Conf. on Fluid and Solid Mechanics, Cambridge

- PERI, D.; DATTOLO, R. (2003), *Systematic Length Variation and Device Positioning Study for a frigate ship accounting propulsion and operability characteristics*, Int. Conf. Ship and Shipping Research, NAV 2003, Ischia
- PERI, D.; CAMPANA, E.F. (2003b), *Multidisciplinary Design Optimization of a Naval Surface Combatant*, J. Ship Research, **47**, 1, 1-12
- PERI, D.; CAMPANA, E.F. (2004), *High-Fidelity models for Multiobjective Global Optimization in Simulation-Based Design*, J. Ship Research
- PARSOPOULOS, K.E.; VRAHATIS, M.N. (2002), *Recent approaches to global optimization problems through Particle Swarm Optimization*, Natural Computing 1: pp.235-306
- SHI, Y.H.; EBERHART, R. C. (1998), *Parameter Selection in Particle Swarm Optimization*, 7th Annual Conf. Evolutionary Programming, San Diego
- SHI, Y.H.; EBERHART, R.C. (1999), *Empirical Study of Particle Swarm Optimization*, Congress on Evolutionary Computing, Vol. III, pp.1945-1950
- SHUBERT, B. (1972), *A Sequential Method Seeking the Global Maximum of a Function*, SIAM J. Numerical Analysis, Vol. 9, pp.379-388
- STATNIKOV, R.B.; MATUSOV, J.B (1995), *Multicriteria Optimization and Engineering*, Chapman & Hall
- TÖRN A.A.; ŽILINSKAS A. (1989), *Global Optimization*, Springer

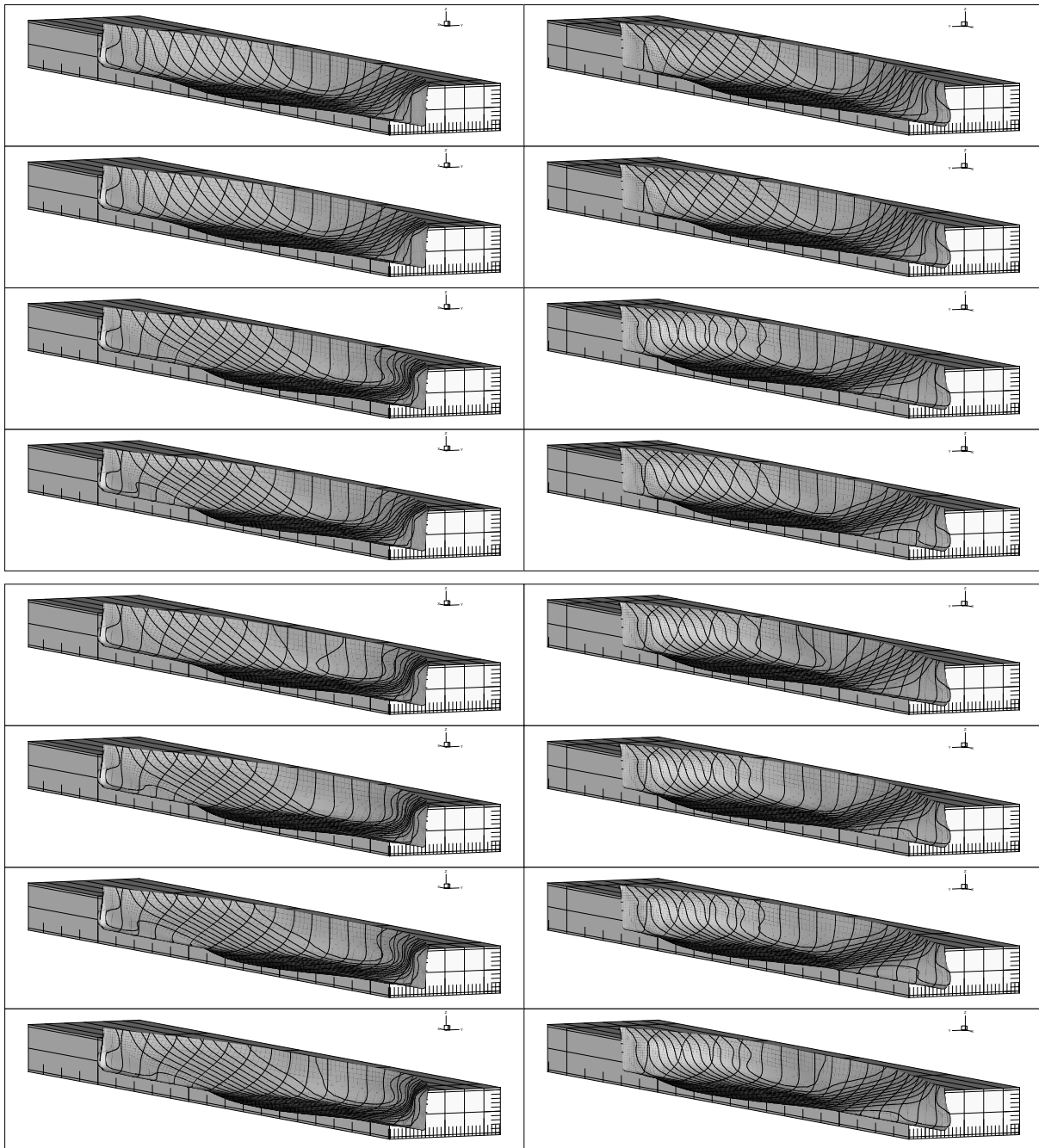


Figure 5: Comparison of the optimized hull shapes obtained by different optimization methods. Each hullform is reported in the front and rear view. From top to bottom: the original hullform, the gradient hullform, the PSO hullform (no group velocity, 12 particles), the SIMPLEX hullform, the PSO hullform (no group velocity, 65 particles), the DRAGO hullform, the PSI hullform, the PSO hullform (group velocity, 24 particles).

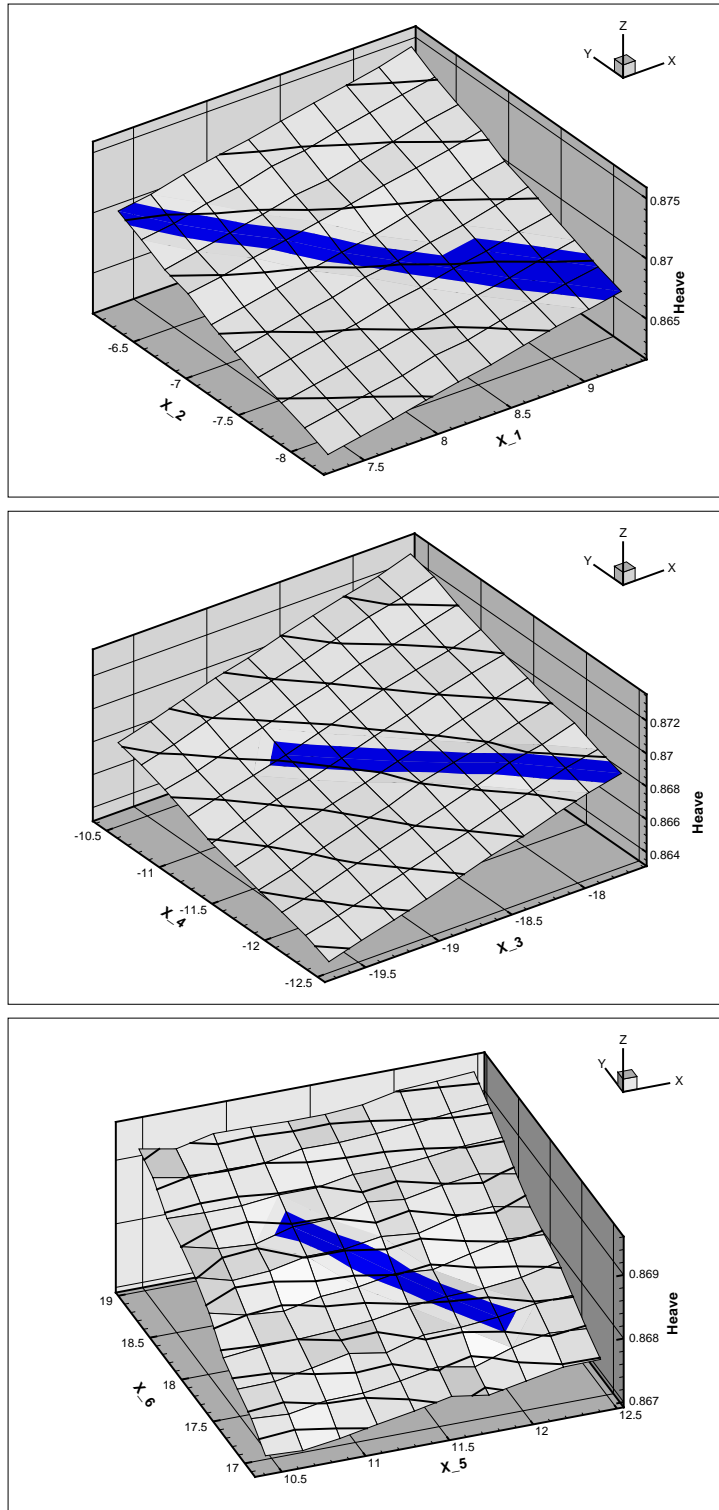


Figure 6: Hyper-plane section of the objective function around the global optimal point. Each picture represents a section of the feasible solution set with an hyper-plane. In the first pictures, design variables n. 3, 4, 5 and 6 are kept constant, while design variables 1 and 2 are regularly varied. On the third axis, objective function is reported. Contour lines indicate the behavior of the objective function. Dark region is the feasible region. The same applies for hyper-planes (3,4) and (5,6).

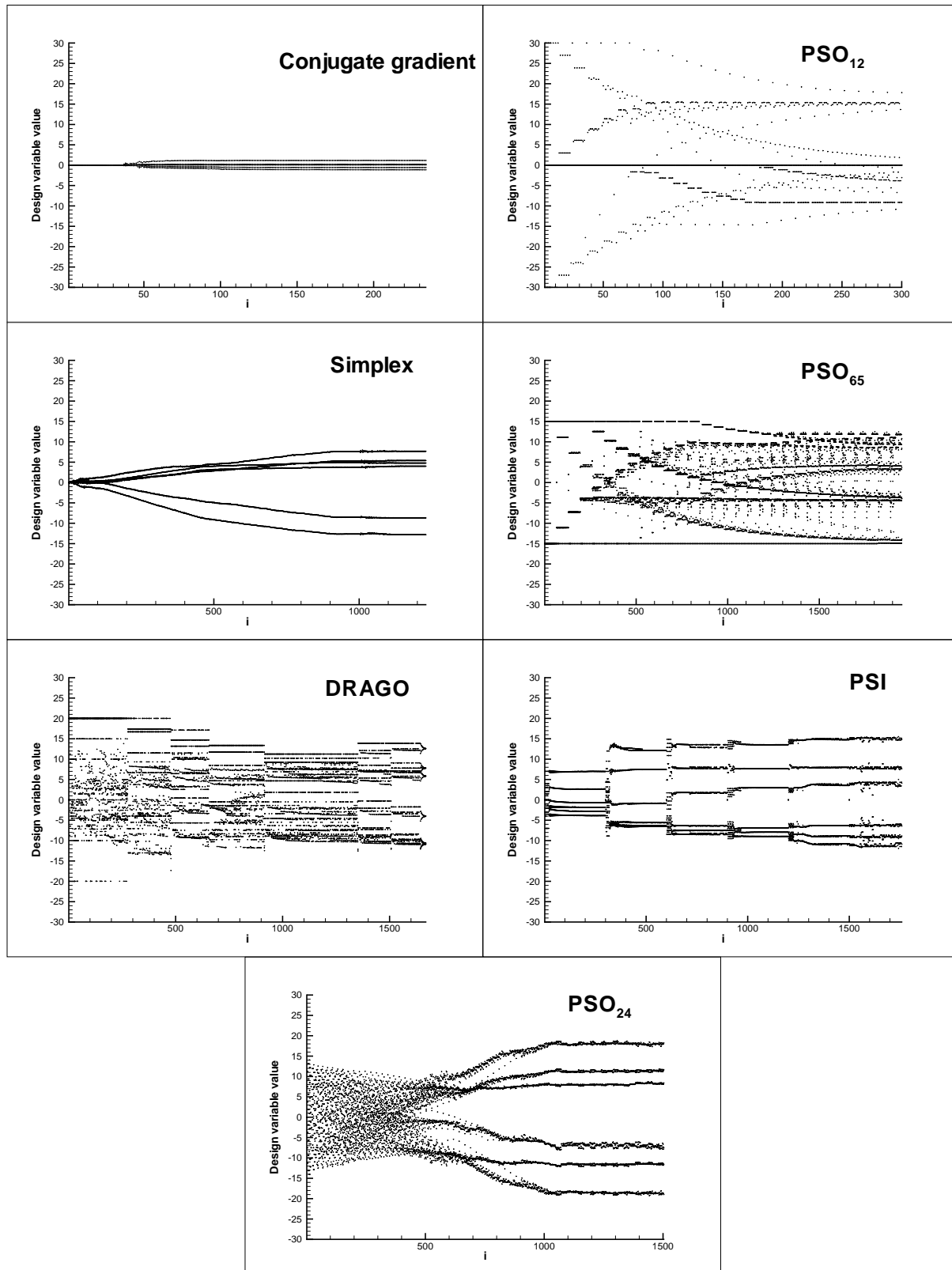


Figure 7: Trend of the design variables as a function of the iteration number for all the tested methods.

Computer Applications and Technologies at Ship Design Group Galati

Virgil Mutu, Ship Design Group Ltd, Galati/Romania, virgil.mutu@vuyk-galati.ro
Ovidiu Ionas, Ship Design Group Ltd, Galati/Romania, o.ionas@vuyk-galati.ro

Abstract

Ship Design Group (SDG) has developed a series of “in-house” computer applications for ship design to support its naval architectural engineering service work. This paper presents briefly software and some applications in simulation and visualisation.

1. Introduction

Ship Design Group Ltd Galati is a design and engineering company founded in 1994, by unifying a group of Romanian private shipbuilding companies under the name Ship Design Group. From 2000 - 2004, Ship Design Group Galati, under the name Vuyk Ship Design Galati, was member of Central Industry Group Holland, shipbuilding service sector. Ship Design Group (SDG) offers design and consulting services, starting with the earlier design stages up to workshop information. Design applications comprise a wide range of ships from smaller service ships (such as tugs, dredgers, offshore supply ships) and pleasure/patrol/rescue craft to multi-purpose and container ships, chemical tankers and ro-ro vessels, bulk carriers and other type of ships, hull and outfitting design.

2. Software

Besides the regular commercial software for basic or detail design (AutoCAD, NUPAS-CADMATIC, etc.) SDG uses a series of software and small applications developed in-house.

SURF with its three main modules:

- **SHAPE** – module for numerical definition and fairing of the hull shape, using advanced numerical methods (Coons, B-spline and NURBS combination), mathematical and graphic methods for checking of surface quality, powerful 3D graphic representation, Fig.1.
- **LINE** – module for cross sections in hull shape using any 2D or 3D section plane, optimization of cross line description and line export to other software.
- **PLATE** – module for numerical definition of seam lines, shell plates expansion, bending information, remarking of stiffness on shell, seam lines offset table. Due to the adjustable expansion method, any plate can be expanded with checking methods for expansion quality. Bending information contain templates, bending and tangent lines on plate and PIN-JIG information. All output data, graphic or text, are in standard format, ready for insertion in the final documents.

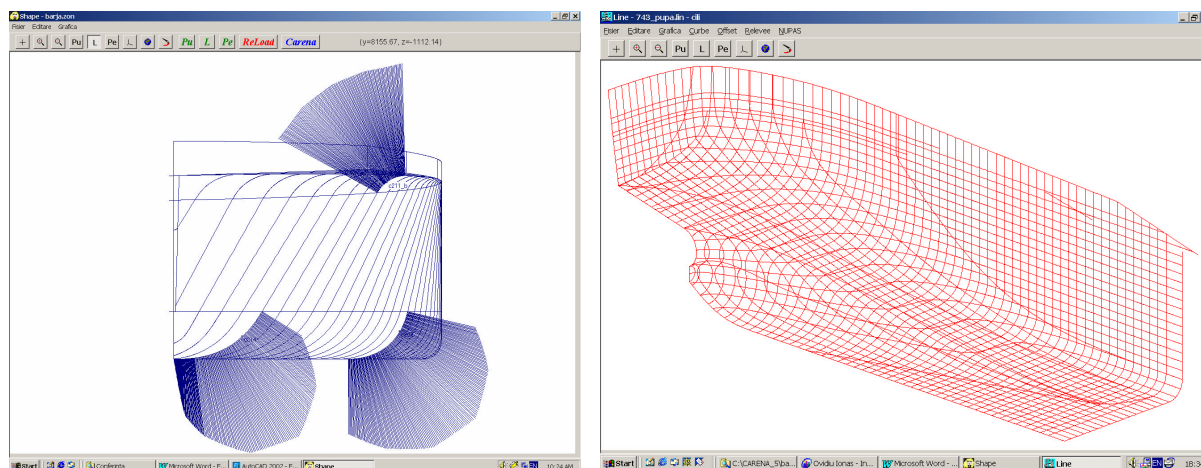


Fig.1: Lines and surface definition using SURF

CARENA is a naval architectural software for: hull and compartment definition, tanks compartment computation, sounding and ullage tables, hydrostatic characteristics, intact and damage stability, longitudinal strength, preparation and report of inclining test, Fig.2. The software has its own module for shape definition or can use shape import from SURF. Graphic and text output is ready for insertion in stability booklet or loading manual.

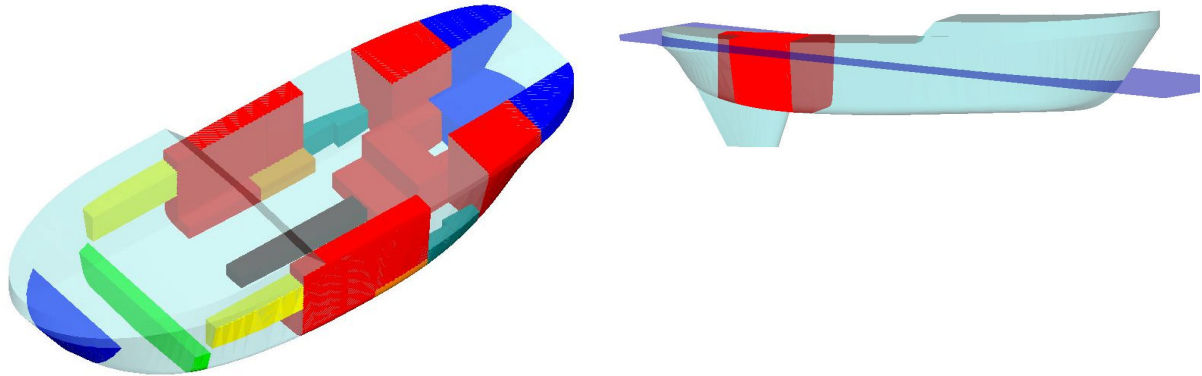


Fig.2: Tanks arrangement and damage stability using CARENA

A mix of integrated systems (specialized task software and in-house software) with suitable interfaces between them seems to be the best way for practical naval applications. SDG uses specialized task software for initial design stage (SURF, CARENA, AutoCAD) and integrated systems for engineering and detail design, some of this software being in-house developed. We have developed link and interfaces between these components to obtain an efficient “integrated software package”, Fig.3.

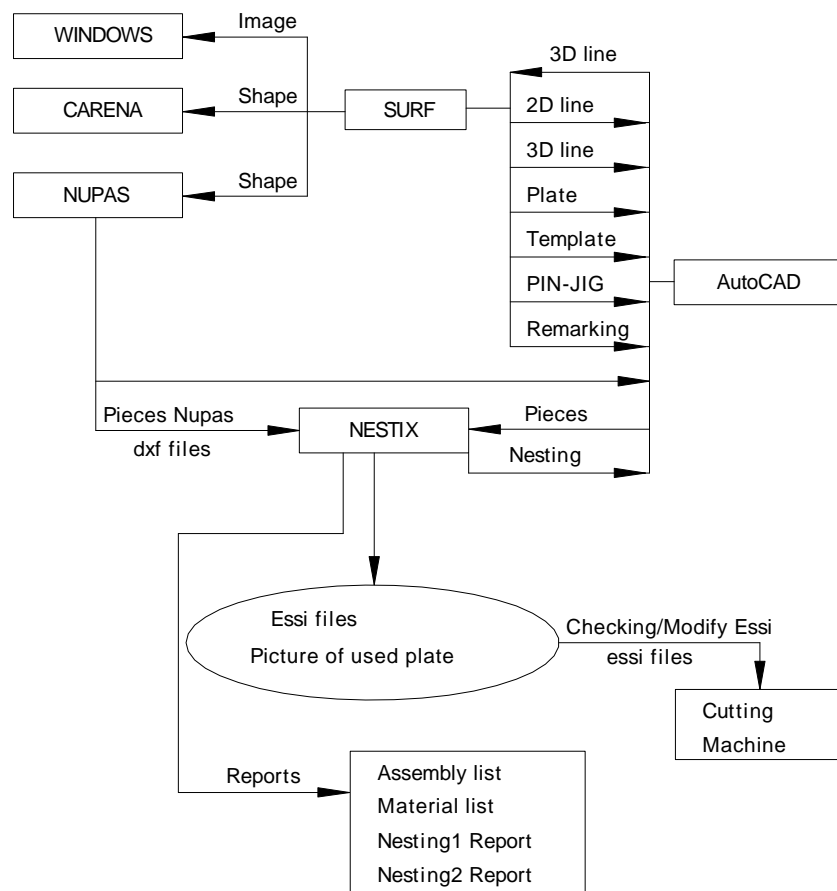


Fig.3: SDG's Software Integration System

3. Selected special projects

Gravitational transversal launching of ships is still used in yards around the world, especially in those located near rivers. In a Romanian yard, the specific conditions of summer 2003 (very low water level of Danube River) made transversal launching very risky. This motivated the development of a computer simulation method. The mathematical model, deriving from the Blagovescensky method was developed following the four principal phases of the transversal launching: slipping, rotation in air, rotation in water, and free-fall. The hydrodynamic and mass coefficients are computed using the database from CARENA for that ship and using special software modules developed for this purpose. The differential equations were integrated in the time domain to give a detailed prediction of the ship motion during launching, Fig.4. Our prediction for five launchings of four different vessels guided decisions for launching safety (avoiding of grounding, avoiding of excessive list, etc.). In some cases the predictions showed the necessity for dredging. Following the advice, the yard managed launching without incidents, Fig.5.

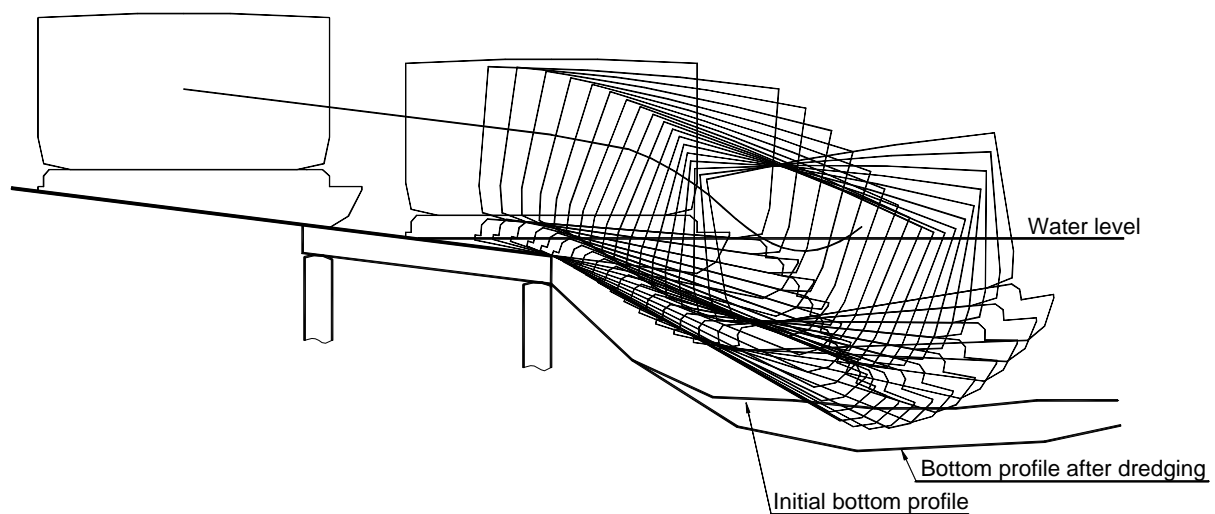


Fig.4: Prediction of launching



Fig.5: Actual launching

Sometimes in ship design it is necessary to predict ship motions in waves. One such situation was the evaluation of inertial forces during the transportation of special cargo (cargo ship carrying two fishing vessels, Fig.6). Another situation in our experience was the evaluation of the environmental conditions for the crew, onboard of a pilot station vessel. These calculations, based on a standard probabilistic approach using sea spectrum, can indicate design solutions and/or operating restrictions.



Fig.6: Cargo vessel carrying “special” cargo

Using 3D modeling and realistic simulation we investigated the loading/unloading operation with AIRBUS containers on a specific vessel. Due to the size of these containers, the container vessel could not be designed and built without investigating first the feasibility of loading and unloading operations. The process began with the 3D modeling of the quay, shore and harbor. In the same time the initial general arrangement of the vessel was converted into a 3D model. The next step was the modeling of the tractor and the trailer (container). Using the maneuvering characteristic from tractor's maker, we developed a mathematical model for generation the trajectories of the tractor-trailer ensemble. Using these trajectories and the 3D models, we simulated loading/unloading of the vessel in different operational conditions and with different types of cargo. These simulations helped to establish the layout of the vessel.

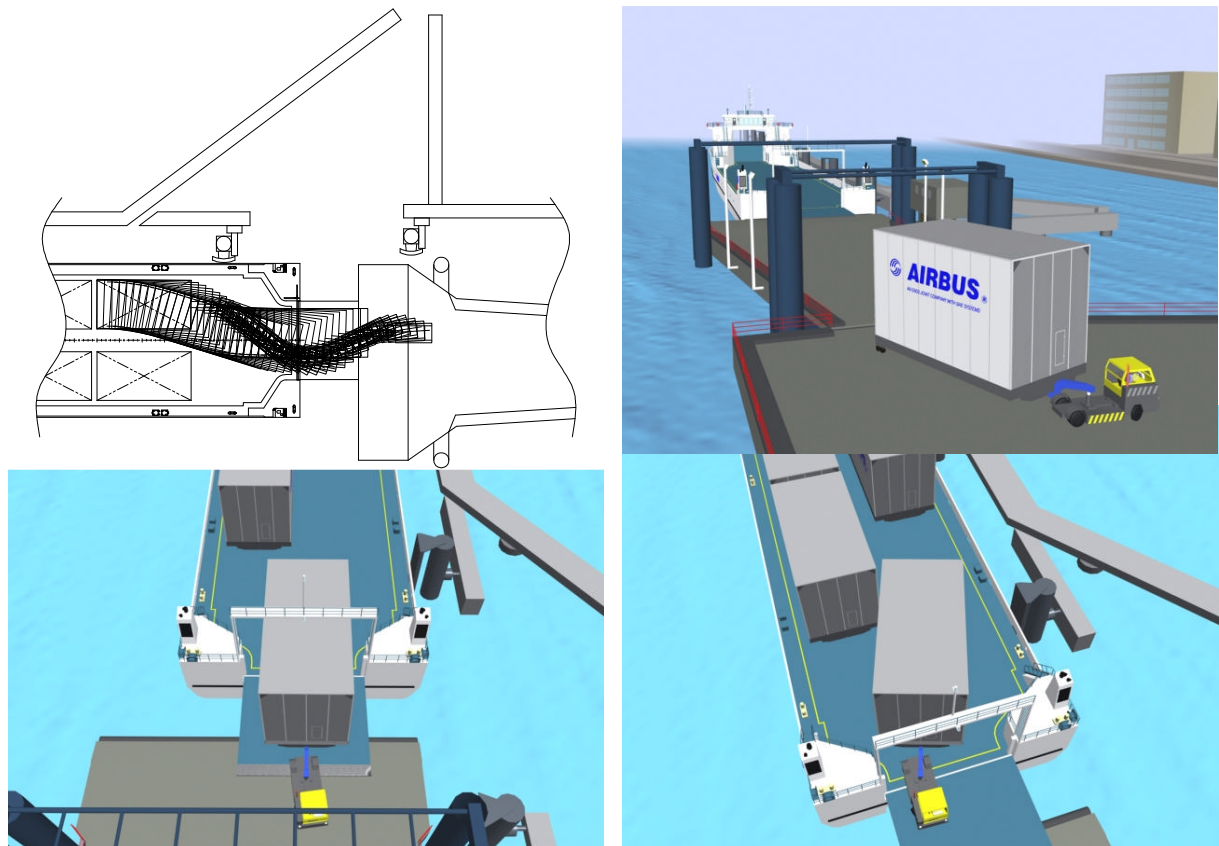


Fig.7: Simulation for loading/unloading of a catamaran container vessel.

The training brig MIRCEA is a special and non-conventional ship, Figs.9 and 10. The ship was modernized and an important task was the pre-stretching of the standing rigging (stays and shrouds). The pre-stretching supposes that the values of the initial applied forces are known and these forces will not allow the appearance of the extreme stress states during sailing. This means values greater than permissible stresses or values that lead to the un-stretching of the standing rigging. The pre-stretching values of the standing rigging were computed using FEM analysis, Fig.11. Each characteristic element of the rigging was modeled according to its shape: the masts and upper masts bowsprit and yards were modeled with beams, the standing rigging with incompressible beams, and the sails with plane shells, Fig.8. Their role is to transmit wind forces. A number of certain navigation cases with different groups of active sails were considered. The equilibrium state was determined iteratively. At each step, the values of the initial pre-stretching forces were modified according to the aim of the calculus. The iterative calculus stopped when in all standing rigging, for all navigation cases, the pre-stretching forces were positive and below the permissible value.

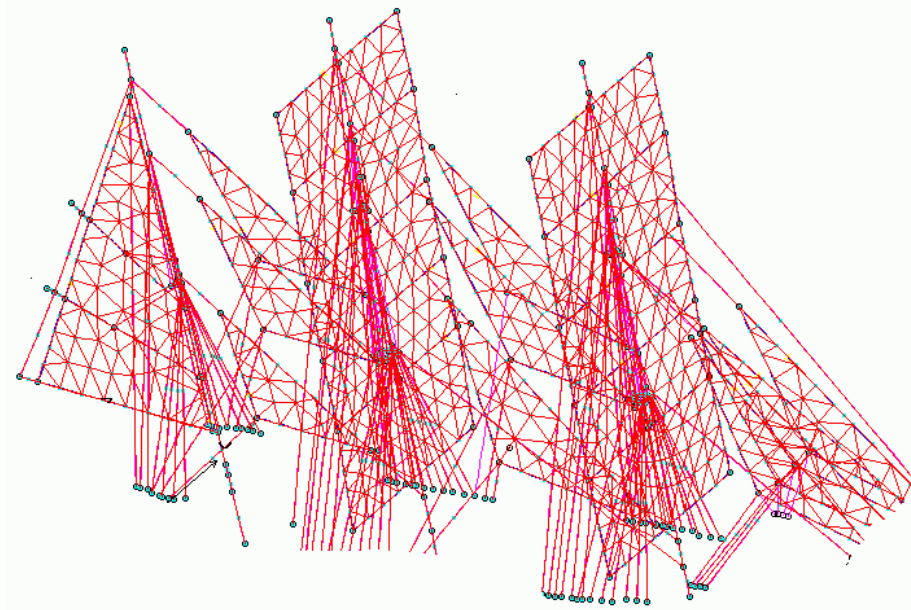


Fig.8: 3D FEM Model



Fig.9: Training brig MIRCEA



Fig.10: Rigging of training brig MIRCEA

In cooperation with Galati University (Department of Naval Architecture), SDG has developed two methods for evaluation of ship resistance in earlier stage of design. The first method is based on CFD simulation. This method is quick but in many cases the results are not enough accurate. The method is

used especially for shape optimization for resistance decrease. Fig.11 compares pressure fields (speed 11 knots) for the initial and modified hulls of a hopper dredger. The second method consists in model tests in the towing tank, Fig.12, after CFD simulation and shape optimization. SDG, together with Galati University, has developed a very quick and efficient technology for model construction and experiments. According to validation with standard ITTC models, the final values of ship resistance are inside an error margin of $\pm 3\%$.

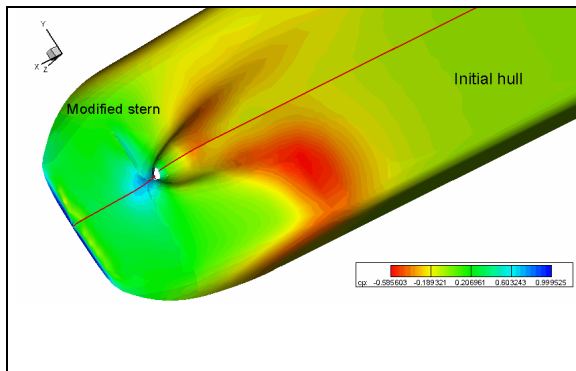


Fig.11: Hull forms improvement for a hopper dredger



Fig.12: University of Galati – Towing tank

References

CHIRICA, I.; IONAS, O.; GIUGLEA, V. (2002), *Specific issues concerning the design of training brig "Mircea"*, TEHNONAV, Constanta

GAVRILESCU, I.; IONAS, O. (1997), *Numerical definition of the hull shape*, The Annals of "Dunarea de Jos" University of Galati, Fascicle XI Shipbuilding.

IONAS, O. (1999), *Contribution in naval architecture based on numerical definition of hull shape*, Ph.D. Thesis, Galati University

IONAS, O.; MARIN, A.; DUMITRIU, O. (2002), *Integrated Design System Surf-AutoCAD-NUPAS-NESTIX*, TEHNONAV, Constanta

LUNGU, A. (2002), *Hull forms improvement for a Hopper Dredger*, Report of Monarch Eng., Galati

MUTU, V.; IONAS, O.; GAVRILESCU, I. (2003), *Surf - An in-house development for a ship hull design software*, COMPIT'03, 2nd Int. EuroConference Computer Applications and Information Technology in the Maritime Industries, COMPIT, Hamburg

An Application Study of Multi-Agent Systems in Multi-Criteria Ship Design Optimisation

Bekir S. Türkmen, The Ship Stability Research Centre, Glasgow/UK bekir.turkmen@na-me.ac.uk

Osman Turan, The Ship Stability Research Centre, Glasgow/UK o.turan@na-me.ac.uk

Abstract

The overall complexity of optimisation problems in ship design results in high computation time requirements. The high computation time requirements for objective function evaluations in a complex design optimisation problem can practically be reduced by either using parallel execution of the objective functions in state-of-the-art super computers or concurrent execution of the objective functions in a distributed environment. In this paper, the distribution of the function evaluations is chosen to improve the performance of the optimisation algorithm. In the proposed distributed decision support environment, the tool for each objective function evaluation is represented as an agent. Agents are the excellent metaphors for devising a distributed environment, because they are able to encapsulate intelligence and tasks modularly as well as taking into account other agents in the environment. However, agents are not the entities, which do all the things for a person without human intervention. The designer of an agent must take into account all the details of the intelligence that the agent requires during its lifecycle. Furthermore, agents need to communicate in an environment, where the naming and role features are solved in order to allow them to co-operate, co-ordinate and be controlled within a certain extent. Multi-agent systems research focuses on three aforementioned topics basically communication, control and collaboration or negotiation. In this paper, the focus is given more on the communication aspects of multi-agent systems, the collaboration and control of the system is addressed by the multi-objective algorithm and the optimisation agent respectively. The optimisation algorithm in the proposed environment is implemented as a multi-objective genetic algorithm because of its capabilities of finding diverse and near Pareto-optimal solutions. The optimisation algorithm is parallelized in the objective function evaluations part of the NSGA II (A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization) and realized by using a multi-agent systems software environment. Two optimisation studies are presented in the paper. In the first optimisation problem, a simple test function is used for the validation of the distributed optimisation algorithm. In the second problem, an internal hull subdivision optimisation problem from the damage stability point of view is implemented and the results and the efficiency of the distributed version are presented. Finally, as one of future directions, the incorporation of the designer's preferences in the optimisation study is discussed.

1. Introduction

Multi-agent systems open the new era of the research topics in computer science and partially in engineering communities. The research on distributed artificial intelligence is shifted to the multi-agent systems, since both address the same concepts. Multi-agent systems are widely used in application areas ranging from control industry to e-commerce. Introduction of the semantic web and XML (Extensible Markup Language) based knowledge systems have also increased the popularity of the agents and in a broader extent to multi-agent systems.

Agents are the entities, in literal meaning, that act or have the power of the authority to act on behalf of its designer or the user. Agents may vary in the literature, from user interface agents to matchmaker agents. The basic and important features of the agents can be listed as autonomy, proactivity and collaboration, especially when we are designing agents to be used for engineering design. An autonomous agent works in a way that it can have self-activation mechanism, for example if we design an agent to check the tolerance of a mechanical system, the agent should be able to have self-activation behaviour to check the tolerance and report to its user, even if the user has not given the stimuli to check except in the involvement of enabling agent at first place. Another example can be

easily given as simple auto-update mechanism, when there is critical operating system patch available, the agent can download and install the patch with user's proper authorization.

The proactive behaviour is a property for an agent to pursue and act according to its goals. There are different proactive approaches in the literature. The RoboCup is one of the main test cases for proactive behaviour. In RoboCup, agents either physical or virtual players, pursue a goal of scoring goals in football game. Planning and team formation methods are applied extensively in proactive collaborative behaviour.

Collaboration is a very important feature of an agent, which also makes an agent differ from an expert system. Collaboration gives the agent to communicate with other agents in the environment for either satisfying its goals or retrieving information in the environment. A very good example of collaborative behaviour can be seen in the nature. Ants randomly wander in the environment for foraging food. When an ant finds food, it carries the food back to its nest, while producing pheromone and leaving the pheromone to its path from the food location to the nest. Randomly wandering other ants sense the traces of the pheromone and follow the path to reach to the food. This simple collaborative behaviour is also investigated in Mars exploration. A reactive robot (There are three types of agent architecture in the literature, reactive, deliberative and hybrid, for further details, *Wooldridge (2002)*) while exploring in the planet collects the magnetic samples and while returning the samples into the base, it drops some magnetic crumbs in order to find its way back to the sample location as well as informing other robots in the environment. There are other agent properties in the literature. However, we omit them in here for the sake of brevity. Interested readers may refer to *Bradshaw (1997)*.

Multi-agent systems combine both above-mentioned and unmentioned agent properties in a framework to make the agents work in a distributed, scalable, dynamic and an open environment. Communication, which is one of the properties of multi-agent systems, can be divided into two categories, semantics and syntax. The syntax addresses the combination of the words in a message among agents. There are few such languages that are proposed in the software community, KQML (Knowledge Query Manipulation Language), *Finin (1994)*, and ACL (Agent Communication Language). The FIPA (Foundation of Physical Agents) organisation created the ACL and encourages the use of ACL wide across in the software systems to enable general communication language among different agent software. Another feature, which may be considered as a more important feature in the communication, is the semantics of the messaging among agents. Semantics can be further divided into two categories, ontology and semantics language. Ontology, unlike the literal meaning, can be explained as the vocabulary of the content of the message in agent communication. Semantics Language is generally based on first order logic and used for information or knowledge queries such as FIPA-SL (FIPA Semantics Language), KIF (Knowledge Interchange Format). See Fig.1 for an example of messaging.

```
(QUERY-REF // FIPA-ACL Communicative Act
:sender ( agent-identifier :name TestAgent@BEKIRLAP:1099/JADE :addresses)
:receiver ( set ( agent-identifier :name CarDeckAgent@BEKIRLAP:1099/JADE ) )
:content "((Have (SideCasing :Side-Casing-Length 0.0 :Side-Casing-Width 0.0 :Side-Casing-Height 0.0)))" // Content Language, 'Have' actually is encapsulated within SL
:language fipa-sl // FIPA Semantics Language
:ontology Deck-Ontology // Side-Casing Length , etc. Concepts and 'Have' Predicate
) // are defined in the Deck Ontology
```

Fig.1: A FIPA-ACL and FIPA-SL based Message between TestAgent and CarDeckAgent

Another property of the multi-agent systems is the control of the system. The control can be introduced into three categories, centralized, federated and autonomous. In a centralized system, one

controller agent co-ordinates task and result sharing generally using blackboard like approach. Tasks are published and results are written into the blackboard. In federated architecture, certain agents (facilitator, matchmaker or middle agents) do the message and task routing. The communication among agents happens via facilitators then facilitators direct the message to corresponding agent(s). The last control mechanism is the autonomous system, in which agent directly communicates with other agents directly without any control.

The last property is the collaboration of the agents in a multi-agent system. Collaboration can be accomplished in many ways, planning, auctions, game theoretic, etc. In this paper, the optimisation agent, a multi-objective genetic algorithm, is used to achieve collaboration or negotiation by means of Pareto-optimality.

Multi-agent systems are also used in engineering design. It is first initiated to enable different knowledge bases to perform in collaborative manner, *Cutkosky (1993)*, and extended to include design management features, *Procura, Goldmann (1996)*. For detailed overview of use of multi-agent systems in engineering design and ship design please refer to *Turkmen and Turan (2003)*, *Shen et al. (2001)*. In *Turkmen and Turan (2003)*, a multi-agent system for ship design decision support for conceptual and preliminary design is proposed, Fig.2. In the system, the multi-criteria decision making tools are added for giving decision support to the designer. The addition of decision support agents are new to previous systems, since in previous systems, the general idea is to have autonomous multi-agent systems in order to design almost without designer's involvement. However, ship design is so highly complex and interrelated that such an autonomous system will be utopia.

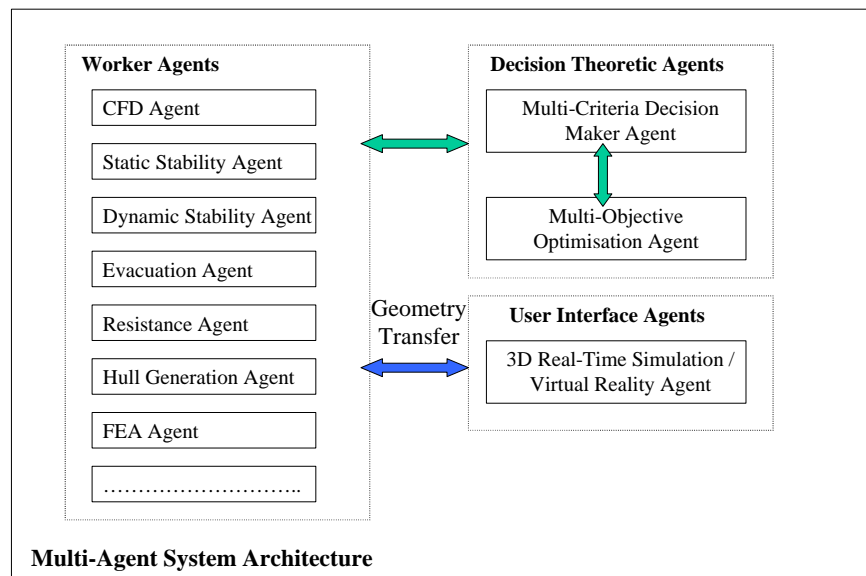


Fig.2: Proposed Multi-Agent System Architecture

In following sections, Multi-objective genetic algorithms, the parallel and distributed NSGAII (A Fast and Elitist Non-Dominated Sorting Genetic Algorithm for Multi Objective Optimization) *Deb et al. (2000)* and the application of the proposed multi-agent system architecture are given for an internal hull subdivision arrangement optimisation problem along with the validation of distributed genetic algorithm with a classical mathematical test function.

2. Multi-Objective Evolutionary Algorithms

The success of finding good Pareto-optimal solutions while maintaining a wide spread of solutions has increased the popularity of evolutionary algorithms in multi-objective optimisation problems. In this study, genetic algorithms are used as an evolutionary algorithm for dealing with multi-objective optimisation problems. Genetic algorithms (GAs) are based on the natural selection, recombination

and mutation concepts for evolving species. Genetic algorithms have the capability of finding global optimum solutions in non-continuous, non-smooth and discrete search domains. The basic definitions for a multi-objective problem are given as following.

Definition 1: Pareto-Dominance Principle

The principle is based on a simple ranking of the solutions by their domination over each other in objective space. Pareto domination can be defined for a minimisation problem as,
For two decision vectors u and v ,

$$\begin{array}{ll} u \text{ dominates } v \ (u \succ v) & \text{iff } f(u) < f(v) \\ u \text{ weakly dominates } v \ (u \succeq v) & \text{iff } f(u) \leq f(v) \\ u \text{ indifferent } v \ (u \sim v) & \text{iff } f(u) \not\leq f(v) \wedge f(v) \not\leq f(u) \end{array}$$

As can be seen from the definitions above u decision vector dominates v if and only if every $f(u)$ objective is less than $f(v)$.

Definition 2: Pareto Optimality

Pareto optimality is actually the optimal solutions, which are not comparable or indifferent from each other in the concept of Pareto dominance. More rigorously, *Zitzler (1998)* a decision vector $x \in X_f$ (feasible set) is said to be non-dominated (non-inferior) regarding a set $A \subseteq X_f$ iff, $\nexists a \in A : (a \succ x)$.

The Pareto optimal solutions of decision vectors are called Pareto-optimal set, non-dominated (non-inferior) set and the corresponding objective vectors form the Pareto-front. In practical terms, a solution is a Pareto-optimal solution, if you cannot gain an improvement in one objective without causing degradation in another objective. We can divide the use of genetic algorithms into two categories for multi-objective problems: Population and Pareto-based genetic algorithms.

2.1.Population-Based Genetic Algorithms

One of the first studies of genetic algorithms in multi-objective optimisation is devised by *Schaffer (1985)*. In his method, Vector Evaluated Genetic Algorithm (VEGA), the simple genetic algorithm is changed only at the selection stage. In the algorithm, every objective is represented by a subpopulation and in every generation, individuals are selected randomly to create a mating pool. After forming a mating pool, the selection is implemented in the mating pool with respect to chosen objective. The same number of individuals for each objective in the selection is preserved for avoiding any bias into search. After selection, individuals are shuffled randomly and then crossover and mutation operators applied. However, VEGA is still affected by the disadvantages of the weighting approach such as not being able to find to non-convex Pareto optimal fronts.

Hajela and Lin (1992) proposed to encode the weightings of the objectives into chromosomes and tried to evolve them via genetic operations. The reason for encoding weightings into chromosomes can be explained as to find best weighting combinations for the aggregation and to generate more solutions with different weighting combinations. In linearly aggregated approaches, the weighting combination is a very important measure to get the decision maker's true preferences. This method is also affected by drawbacks of the weighted aggregation method, since it uses the weighted sum approach in its core.

2.2.Pareto-based Genetic Algorithms

Most of the algorithms in this section use Pareto-based fitness assignment proposed by *Goldberg, (1989)*. In Goldberg's proposal, firstly non-dominated front is found, and then all the individuals on that front are assigned to a same dummy fitness. For the second front, the first front is ignored and the

dummy fitness values for those individuals in the second front are assigned but this time the dummy fitness values are reduced in order to preserve the non-dominance property of the first front.

Fonseca and Fleming (1993) proposed a different scheme for the fitness assignment, an individual's fitness assigned depending on how many individuals in the population it is dominated by. As a result all the non-dominated solutions are assigned to the same fitness, which is zero.

Horn and Nafpliotis (1993) approached the Pareto-based fitness assignment in a different way. The proposed method NPGA (Niche Pareto Genetic Algorithm) used the tournament selection operator. In NPGA, two individuals are chosen randomly from the population, such as A and B. Moreover, t_{dom} individuals from the current population ($t_{dom} > 3$) are again randomly chosen. The algorithm begins with the comparison of selected individuals A and B with the selected t_{dom} individuals in the tournament set. If B is dominated but A is not, A is selected. If both are dominated or non-dominated, the fitness sharing is implemented. The fitness sharing uses continuously updated sharing, in which niche counts are calculated not by using current population, but rather partly filled next parent population. In other words, the distance metric for the sharing is implemented on the selected individuals for the next parent population.

Srinivas and Deb (1995) applied Goldberg's fitness assignment approach directly with Non-dominated Sorting Genetic Algorithm (NSGA). The fitness sharing is also applied in this method in each front in order to improve the wide spread of solutions. However, the method has disadvantages for its computational complexity.

While the above-mentioned methods are all dependent on a sharing parameter s_{share} , for fitness sharing, the following methods do not need s_{share} but they use the similar ideology which is fitness sharing.

SPEA, *Zitzler (1998)*, (Strength Pareto Evolutionary Algorithms) is another multi-objective genetic algorithm method. SPEA uses an external set for creating a non-dominated set and adds this external set into selection and fitness sharing process. In SPEA, fitness values of the individuals are defined by its strength. Fitness sharing is implemented by new Pareto-based niching method. In Pareto-based niching method, an individual's strength can be reduced or improved by Pareto dominance principle instead of relative distance among individuals. The algorithm also uses clustering analysis to reduce the number of the non-dominated individuals in the external set not to slow down and unbalance the algorithm's performance.

PAES (Pareto Archived Evolution Strategy) is proposed by *Knowles and Corne (1999)*. In this method, one parent and one children evolution strategy is used. "The child is compared with the parent. If the child dominates the parent, the child is accepted as the next parent and the iteration continues. If the child is dominated then the parent is mutated and the iteration is continued. If they cannot dominate each other then the child is compared with an archived external set created by the best solutions. The algorithm then accepts or ignores the child with respect to crowding property of the child", *Deb et al. (2000)*. Details of the algorithm are given in *Knowles and Corne (1999)*.

NSGAI, a fast and elitist multi objective genetic algorithm, is proposed by *Deb et. al (2000)* and in the algorithm NSGA's fitness assignment is improved with better book keeping algorithm. The algorithm also uses the crowding distance parameter to keep diversity in the population. It also defines the constraint-dominance principle in order to work better with constrained optimisation problems. There are other algorithms based on multi-objective genetic algorithms, however, they are omitted for brevity.

3. Agent Oriented Programming and Distributed NSGAI

Agent oriented programming differs from object oriented programming in a few ways. Agents are built up from behaviours and those behaviours represent mental states of the agents. Behaviours also

govern agents' interaction with the environment and with other agents. For an agent to accomplish all sorts of different behaviours in timely manner, the system should be able to handle different events asynchronously. In the implementation of agents and multi-agent systems in this paper, JADE, *Jade* (2004)] is used for multi-agent software platform. JADE is an agent platform, which comply with FIPA tests for interoperability. JADE agents can be embedded into web pages, as applets or JAVA Server Pages, or they can reside in the local host. JADE can also work in connection with JESS, *Jess* (2003) and Protégé, *Protégé* (2004), for ontology support.

The optimisation algorithm is distributed and parallelized in the objective evaluations and update parts of NSGAII algorithm. Before going into detail of the software design issues, detailed explanation of NSGAII will be necessary for the background information.

3.1. The NSGA II Algorithm

We can divide NSGA II into three sections. The first one is non-dominated sorting for fitness assignment. The second one, is fitness sharing (crowding distance assignment) and the last section is the binary tournament selection with respect to fitness and crowding distance values.

Non-dominated sorting in NSGA II is a fast and modified version of *Srinivas and Deb's* (1995) proposal NSGA. In this sorting procedure, non-dominated individuals are found in the population with Pareto-dominance principle and assigned to rank zero, then they are ignored to find second front and so on.

Crowding distance assignment is used in order to penalize the individuals that are closer to each other according to fitness sharing. The assignment works in the following way; individuals are sorted with respect to each objective for the optimisation problem in ascending order. The border individuals, the individuals having lower and upper values for each objective function, are assigned to an arbitrary big value. Assigning big values to border individuals is applied in order to preserve the border individuals to find wide spread Pareto-front. Crowding distance values for every objective are then calculated and summed to find a single value for the individual's crowding distance. This procedure is applied for each front.

The third main part is the crowding distance selection. This selection method is a modified version of the binary tournament selection. The only difference in this selection method is that, the method compares crowding distance values of two selected individuals as well as comparing fitness values (ranks, in this case), when two individuals are in the same rank.

NSGA II is an elitist multi-objective genetic algorithm. The NSGA II implements elitism by combining parent and child populations for every generation. The selection of the individuals to create new parent population from two sets is realized by sorting the last front with respect to crowding distance in descending order and selecting the individuals having higher crowding distance values.

3.2. Distributed NSGAII

As mentioned before, for the objective function evaluations and updates we distribute the computational load to other computers in the environment. In order to achieve this, two modifications have been made to the current local version of NSGAII algorithm. The first modification is needed to devise a behaviour for allowing us to distribute the computation among agents and the second modification is the basic loop of generation module in the local version.

3.2.1. The Behaviour Design for Distribution of Objective and Constraint Evaluations

A behaviour must be devised to send the job to the other agents, and wait for the results for certain amount time and inform optimisation agent which embeds NSGAII algorithm. Thankfully, JADE has built-in behaviours for achieving the above-mentioned behaviour. The behaviour in the JADE is

called AchieveREInitiator/Responder which are defined in order to implement FIPA-Request like interaction protocols, such as FIPA-Request, FIPA-Query. This behaviour is used to achieve rational effect, the expected effect of the action. A protocol is used in here in order to keep track of the status of rational effect's achievement, *Bellifemine (2003)*. In the current implementation, the optimiser agent uses the initiator behaviour, but other worker agents use simple behaviour of JADE, which compares the communicative act and sends the result, Fig.3

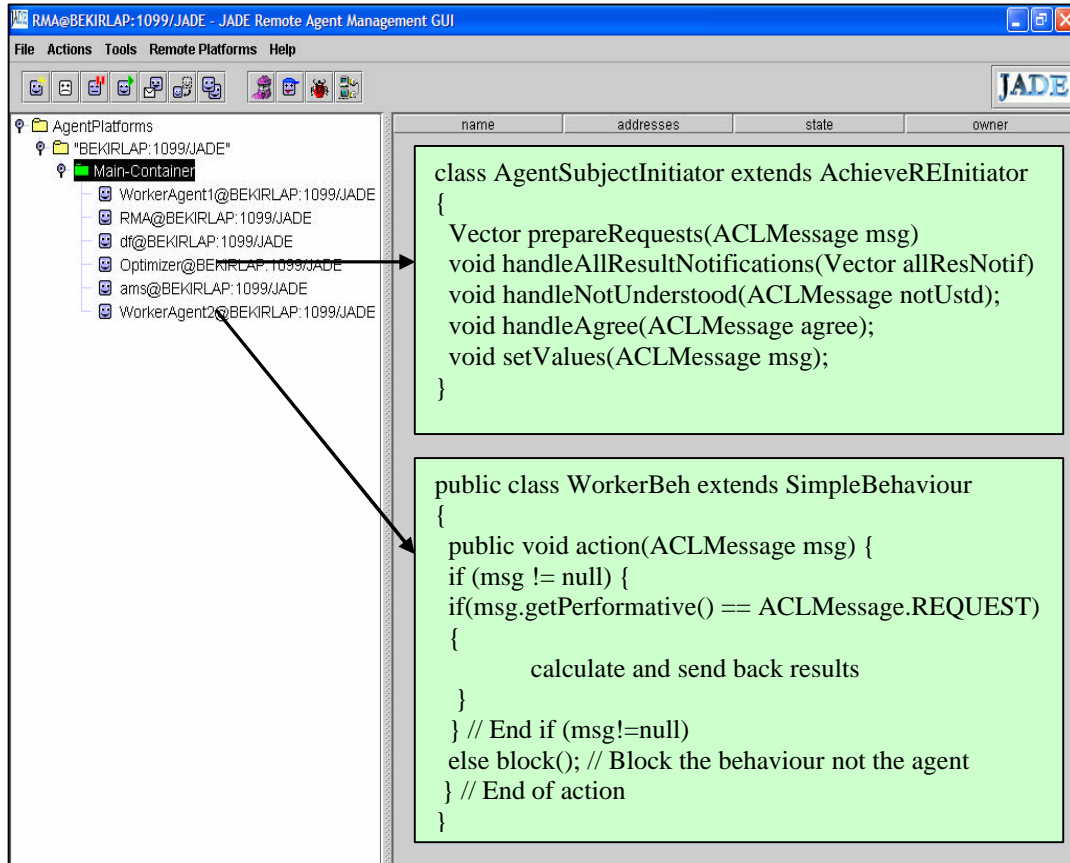


Fig.3: Description of Agent Behaviours and JADE GUI

Other types of behaviour, which need to be added to the agent's behaviour queue are the update and evaluate behaviours for the optimisation algorithm in order to set the objective and constraint function fields of the individuals in the population.

3.2.2. Modified Generation Method for NSGAI

The generation needs to be divided into groups of behaviours and pushed into agent's behaviour queue in order to run the algorithm. In JADE, when the behaviour is done and finished, it is also removed from the behaviour's queue. This results in adding behaviours in every generation. The NSGAI is divided into five main groups, evaluation, first generation, update, second generation and update. The generation behaviour is devised as a sequential behaviour loop. In this sequential behaviour, evaluate and update behaviours are added as parallel behaviours, Fig.4.

There are two missing implementation details in this distributed algorithm. The first one is related to the task sharing, instead of AchieveREInitiator, the more advanced version of Initiator, Contract Net protocol may have been used, and secondly the optimisation algorithm waits for results from every agent to proceed (Granularity problem). Those missing parts can be easily implemented and they will be further detailed in the future research part.


```

// Create generation behaviour
SequentialBehaviour gen=new SequentialBehaviour(myAgent)
{
    onEnd() {
        if(genCounter<nGens-1) {++genCounter; ...}
        else { // clean-up and print out results
        }
    }
}
// Evaluate Objective & Constraint Evaluations
addEvaluateBehaviours(); // Parallel Behaviours added to the gen, in a sequential
behaviour loop
// First Generation
gen.addSubBehaviour(new RunFirstGenerationBehaviour()); // OneShotBehaviour
// Update ChildInds
addEvaluatebehaviours();
// Run Second Generation
// Loop Through Until Generation Counter Exceeded The Max. Limit
runAndUpdateSecondGeneration(); // Add generation behaviour to the agent's queue
//myAgent.addBehaviour(gen);

```

Fig.4: Generation Behaviour for NSGA II Algorithm

3.3. The Test of The Algorithm with a Classical Test Function

In order to validate the code, the classical Schaffer test function is used. The local code has been already tested in various test functions for validation. The Schaffer Test Function F2, *Schaffer et al. (1985)*, is given below.

$$\begin{aligned}
 \text{Min. } f_1(\mathbf{x}) &= x^2, \\
 \text{Min. } f_2(\mathbf{x}) &= (x-2)^2, \\
 -1000 < x < 1000
 \end{aligned} \tag{1}$$

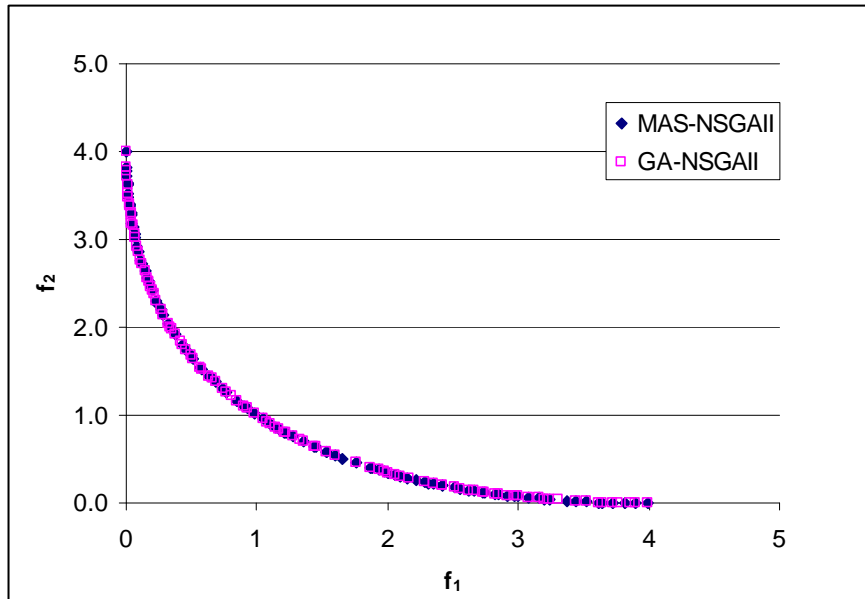


Fig.5: Comparison of Distributed Version with Local Version

The test run parameters for this optimisation problem are as follows; The population parameters for this problem are; Probability of single point crossover, p_c , 0.9, probability of random mutation, p_m , 0.01, and number of generations, nGens, 250 and the population size, nInds, 100. Two agents are created in order to calculate above functions. The agents are executed locally in order to measure the performance. Fig.5 compares results showing that the algorithm works as expected, however, the time took for the calculation in distributed version is increased. (That is already expected, since agents and JADE Agent platform will cause the algorithm to run slower than its local counterpart)

That is also important to note that, the validation of the code is just to ensure that the distributed code works the same way as its local version does. Although, the distributed code uses the local code segments, the behaviour management must be carefully investigated in the software. JADE provides two tools for debugging agent applications, introspector and sniffer agents. The sniffer agent is used in the implementation of developed software for debugging.

4. Application Study: Internal Hull Subdivision of a Ro-Ro Passenger (ROPAX) Vessel

The optimisation problem for applying the distributed algorithm in ship design is an internal hull subdivision problem for a Ro-Ro Passenger ship (ROPAX). The tragic accidents of the Herald of Free Enterprise in 1987 and the Estonia in 1994 initiated a significant surge of research related to the capsizing of Passenger Ro-Ro type of ships, which have a very large undivided car deck. Flooding of that undivided car deck even with a small amount of water caused rapid capsize of the ships and loss of significant number of lives as experienced with above-mentioned tragic accidents. This effort culminated in significant developments that helped the ferry industry to raise safety levels to demanding new heights. New regulations require all the ROPAX Vessels to comply with SOLAS' 90, *SOLAS (2001)*, standards as well as new water on deck standards known as the Stockholm Agreement, which determines the limiting wave height at which ROPAX vessel survives in damaged condition. In response to the strict new regulations, the shipping industry is demanding new modern Passenger Ro-Ro designs with very high standards cost effectively.

Furthermore, due to recent European Union (EU) regulations, which banned the tax-free sales during the journeys between EU countries, forced designers and operators to look for design and route alternatives to meet customer demands in line with high safety standards as well as cost effectiveness. The end of tax-free sales created a surplus store space in those designs, which cannot be used for passenger accommodation due to safety regulations. The main aim of this case study is to maximise the survivability and stability standards while improving the cargo capacity by introducing new design alternatives. The internal arrangement of the hull is a very important factor for a ship's damage stability and survivability, especially when damage occurred. The optimisation problem is therefore limited to arrangement of transverse bulkheads, car deck height, lower hold height, and lastly side casing, which is known to be important parameter for increasing stability and survivability, Fig.6. The problem described here is a successive work of *Olcer et al. (2003)* and objectives of this optimisation problem are given in Table I. The optimisation variables are also given in Appendix A.1.

Table I: Optimisation Objectives

No	Objectives	Type	Description
1	H_s value	Maximisation	for the worst two compartment damage case
2	KG limiting value	Maximisation	for the worst two compartment damage case
3	Cargo capacity	Maximisation	expressed in car lanes, lorry lanes kept constant

There are three objectives in this study. The first objective is the maximisation of H_s (Significant wave height), which is a measure used to assess survivability of ships in waves. Static Equivalent Method, *Vassalos et al. (1996)*, *Tagg and Tuzcu (2002)*, is deployed here in order to calculate H_s .

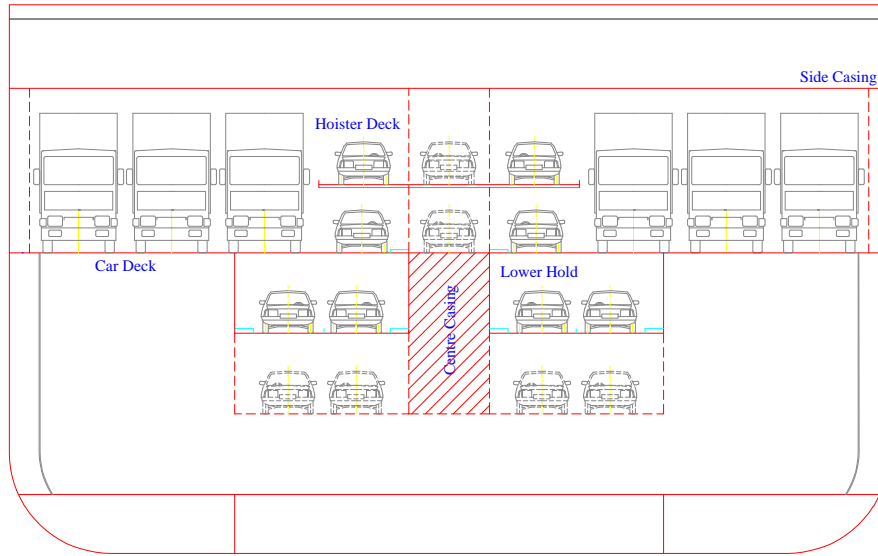


Fig.6: ROPAX Ship Sectional Representation

Another objective is to achieve the right stability standards by achieving limiting KG (vertical centre of gravity) as high as possible. Limiting KG is a way of expressing allowable loading condition of a ship. It is a crucial fact that limiting KG must be higher than the KG value for a loading condition (operational KG) in order to achieve enough stability. Operational KG is a very important parameter for a ship, since it tends to increase during the life cycle of the ship and that causes changes in loading condition. The higher limiting KG values enable designers to make design changes during the life cycle of a ship. The third objective is earning capacity of the ship, which is defined as the number of car lanes in the ROPAX ship.

There are also constraints in this optimisation problem in order to comply with SOLAS'90 Regulations, such as minimum GZ Area, and maximum heel. The minimum bulkhead distance for two adjacent bulkheads must be greater than $0.03.L_s + 3.0\text{m}$. We applied this approach in order to comply with SOLAS'90 regulations with no additional calculation cost. L_s is the subdivision length defined in the SOLAS' 90 regulations. Another important constraint in the case study is the operational constraint, which is used to satisfy operational KG requirements in the life cycle of the ship. The tabulated form of all constraints is given in Appendix A.2.

The constraint approach for minimum bulkhead distance is modified in order to make the search more robust. The approach applied here sums all violating bulkhead distances for an individual (string or inputs for optimisation problem in genetic algorithms) and NSGAI with constraint violation penalizes the individuals, which violate the bulkhead distance constraint more.

4.1. Representation of Results

Two different runs are carried out, the first one is the local version and the second is the one with multi-agent systems based distributed genetic algorithm. Same parameters are used in both runs. The population size, number of individuals in the population, is set to 80 and the number of generations is set to 32. Real coding is used as a chromosome representation. Although, the real coding of chromosome representation is used, the crossover and mutation operators are selected as one-point crossover and random mutation. The reason for using crossover and mutation operators, which are not generally suitable for real coding, was due to the problem's combinatorial property. The transverse bulkhead positions along with side-casing width and lower hold height are discrete and that is why, the problem can be seen as a combinatorial problem. One can argue that, using multi-point crossover may perform better in this kind of problems by better preserving building blocks. However, the aim is here to show the multi-agent systems based distributed genetic algorithm, and also we strongly

believe that, even if the multi-point crossover used, the best individual found would not differ too much, due to problem's nature.

In Table II, the best individuals found in both runs are shown. The difference between the best-found individuals in both runs is due to the random features of the genetic algorithms. The selection of the design will of course be the best design found from MAS-NSGAI.

Table II: The Original Design and Found Best Designs

No	Optimisation Variables	Original Design	GA-NSGAI	MAS-NSGAI
1	Car deck height	9.7m	9.9	9.9 m
2	Side-casing width	No side-casing	1	1 m
3	Lower-hold height (from car deck)	2.6m	5.2	5.2 m
Watertight Transverse Bulkheads		In frame numbers	In frame numbers	
4	Transverse bulkhead 02	27	29	26
5	Transverse bulkhead 03	39	41	38
6	Transverse bulkhead 04	51	52	49
7	Transverse bulkhead 05	63	65	65
8	Transverse bulkhead 06	81	83	83
9	Transverse bulkhead 07	99	97	98
10	Transverse bulkhead 08	117	115	115
11	Transverse bulkhead 09	129	127	127
12	Transverse bulkhead 10	141	143	140
13	Transverse bulkhead 11	153	155	151
14	Transverse bulkhead 12	165	166	164
15	Transverse bulkhead 13	177	177	179
16	Transverse bulkhead 14	189	190	191
Performance scores of designs				
1	Cargo capacity expressed in car lanes	8 (abt.1033 m.)	14 (abt.1450) m	14 (abt.1450) m.
2	H _s value	4.641 m	5.443 m	5.908 m
3	KG limiting value	13.845 m	14.048 m	14.044 m

The time required for the algorithm is reduced about 40 % percent. The question may arise why the reduction was not 50 %, the answer to this question is two folds. The first one is that the algorithm runs slower than its counterpart due to communication and Jade's involvement, and secondly, the distributed algorithm has a granularity problem as mentioned before.

5. Discussions and Conclusions

The paper represented a multi-agent systems based distributed genetic algorithm in a master/slave parallel genetic algorithm. The algorithm can be further improved with investigating on island model genetic algorithms to reduce the communication needs, however there could be other problems such as selecting migration rate or how many times the migration will be implemented. The algorithm has so called granularity problem, but we strongly believe that the algorithm may be improved with better scheduling with little effort.

The use of multi-agent systems can also be further improved by using contract net approach, for task and result sharing, and furthermore, Directory Facilitator (In JADE, agents can register themselves to the directory facilitator, with their service description, the semantics language and so on, in other words, the directory facilitator works as a yellow page service) may be used to find the agents in the environment in order to make optimiser agent more dynamic in changing environments.

The selection of the best individual is performed by designer's preferences, to select among almost 10~ 15 near Pareto-optimal solutions. The designer's preferences are valuable ingredient for the optimisation especially for engineering design problems, having time and resource bounds. The introduction of designer's preferences may have been introduced in three categories. The first category, a priori, such as the weighted aggregation techniques or posteriori as the way it has been introduced for the solution of this problem, or progressive, the algorithm can be guided during the optimisation with changing designer's preferences. The third approach, progressive designer's involvement, is obviously the best choice among three, however when there is an extensive need for designer's involvement may make the method discouraging and tedious to use. An intelligent agent which will be acting on behalf of designer in order to guide search very important asset to the solution of the problem.

In conclusion, a multi-agent systems based genetic algorithm is implemented and implementation details are outlined. The algorithm performed well both in mathematical test function as well as ship design problem. The original ROPAX design is improved for survivability and operational parameters, operational KG and Car- lane length. Increasing the lower hold height has also enabled to load truck's into the lower hold, which enables the ship to work on different load conditions.

References

BELLIFEMINE, F.; GIOVANNI C.; TIZIANA T.; GIOVANNI R. (2003), *Jade Programmer's Guide*, <http://sharon.cselt.it/projects/jade/>

BRADSHAW, J. M., Editor (1997), *Software Agents*, The AIAA Press

CUTKOSKY, M.; ENGELMORE, R.; FIKES, R.; GRUBER, T.; GENESERETH, M.; MARK, W. (1993), *PACT: An Experiment in Integrating Concurrent Engineering Systems* IEEE Computer, special issue on Computer Supported Concurrent Engineering, 26/1, pp.28-37

DEB, K.; AGRAWAL, S.; PRATAB, A.; MEYARIVAN T. (2000), A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA II, KanGAL Report, 200001, Indian Institute of Technology, Kanpur, India

FININ, T.; FRITZON, R.; MCKAY, D.; R MCENTIRE. (1994), *KQML as an Agent Communication Language*, Proceedings of 3rd Int. Conf. on Information and Knowledge Management, ACM Press

FONSECA, C. M.; FLEMING, P. J. (1993), *Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization*, Proceedings of The Fifth International Conference On Genetic Algorithms, Schaffer, D. Editor, Morgan Kaufmann, San Mateo pp. 416-423

GOLDBERG, D. E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, Reading, Massachusetts

GOLDMANN, S. (1996), *Procura: A Project Management Model of Concurrent Planning and Design*, Proc. WETICE-96, IEEE Computer Soc. Press, pp. 177-183

HAJELA, P.; LIN, C-Y. (1992), *Genetic search strategies in multicriterion optimal design*, Structural Optimization, 4, 99-107

HORN, J. and NAFPLIOTIS, N. (1993), *Multiobjective Optimization using the Niche Pareto Genetic Algorithm*, Technical Report, IlliGAL Report, 93005, University of Illinois, USA

JADE (2004), *Java Agent Development Environment*, <http://sharon.cselt.it/projects/jade/>

JESS (2003), *The Expert System Shell for the Java Platform*, <http://herzberg.ca.sandia.gov/jess/>

KNOWLES, J.; CORNE, D. (1999), *The Pareto archived evolution strategy: A new baseline algorithm for multiobjective optimisation*, Proceedings of the 1999 Congress on Evolutionary Computation, pp. 98-105

OLCER, A.; TUZCU, C.; TURAN, O. (2003), *Internal Hull Subdivision Optimisation of Ro-Ro Vessels in Multiple Criteria Decision Making Environment*, Proceedings of The 8th International Marine Design Conference, pp. 339-351

PROTÉGÉ (2004), *Ontology Editor and Knowledge Acquisition System*, <http://protege.stanford.edu/>

SCHAFFER, J. D. (1985), *Multiple objective optimization with vector evaluated genetic algorithms*, In J. Grefenstette (Ed.), Proceedings of an International Conference on Genetic Algorithms and Their Applications, Pittsburgh, PA, pp. 93-100

SHEN, W.; NORRIE, D.H.; BARTHES, A.J. (2001), *Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing*, Taylor&Francis

SOLAS Consolidated Edition (2001), International Maritime Organization, London

SRIVINAS, N.; DEB, K. (1995), *Multi-objective function optimisation using non-dominated sorting genetic algorithms*, Evolutionary Computation, 2, pp. 221-248

TAGG, R.; TUZCU, C. (2002), *A Performance-based Assessment of the Survival of Damaged Ships - Final Outcome of the EU Research Project HARDER*, Proceedings of the 6th International Ship Stability Workshop, Webb Institute

TURKMEN, B. S.; TURAN O. (2003), *Multi-Agent Systems in Ship Design*, Second Int. EuroConf. on Computer and IT Applications in the Maritime Industries COMPIT'03, Hamburg, pp. 459-472

VASSALOS, D.; PAWLOWSKI, M.; TURAN, O. (1996), *A theoretical investigation on the capsizing resistance of passenger Ro/Ro vessels and proposal of survival criteria*, Final Report, Task 5, The NorthWest European R&D Project

WOOLDRIDGE, M. (2002), *An Introduction to MultiAgent Systems*, John Wiley & Sons

ZITZLER, E. (1998), *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*, PhD Thesis, Swiss Federal Institute of Technology, Zurich

Appendix A

Table A-I: Optimisation Variables

No	Variables	Bounds		Increment
		Lower	Upper	
1	Car deck height	9.6m	9.9m	0.025m
2	Side-casing width	1m	2m	0.5m
3	Lower-hold height (from car deck)	2.6m	5.2m	-
4	Transverse Bulkhead 02	25	29	1
5	Transverse Bulkhead 03	37	41	1
6	Transverse Bulkhead 04	49	53	1
7	Transverse Bulkhead 05	61	65	1
8	Transverse Bulkhead 06	79	83	1
9	Transverse Bulkhead 07	97	101	1
10	Transverse Bulkhead 08	115	119	1
11	Transverse Bulkhead 09	127	131	1
12	Transverse Bulkhead 10	139	143	1
13	Transverse Bulkhead 11	151	155	1
14	Transverse Bulkhead 12	163	167	1
15	Transverse Bulkhead 13	175	179	1
16	Transverse Bulkhead 14	187	191	1
Bounds for transverse bulkheads are given in frame numbers				

Table A-II: Optimisation Constraints

No	Constraints	Requirements
1	Range	Range of positive stability minimum of 15 degrees
2	Min. GZ Area	Minimum area of GZ-curve at least 0.015mrad
3	Maximum GZ Taking into account the greatest of those heeling moments	<ol style="list-style-type: none"> 1. The crowding of all passengers towards one side 2. the launching of all fully loaded davit-launched survival craft on one side 3. due to wind pressure $GZ \text{ (in metres)} = \frac{\text{heelingmoment}}{\text{displacement}} + 0.04$ <p>However, in no case is this righting lever to be less than 0.1 m.</p>
4	Maximum Heel	Maximum static heel not more than 12degrees (two adjacent compartment)
5	Minimum GM	Minimum GM (Metacentric Height) at least 0.05m
6	Margin Line	Margin line should not be immersed
7	Bulkhead distance	Minimum bulkhead distance ($0.03 \cdot L_{bp} + 3m$)
8	KG Operational	Limiting KG greater or equal to 1.005 of Operational KG

A New Collaborative Engineering Management Tool for Shipbuilding

Rafael de Góngora, SENER Ingeniería y Sistemas SA, Madrid/Spain, rafael.gongora@sener.es

Abstract

Once the use of 3D CAD/CAM tools has been commonly accepted in the shipbuilding industry, it is necessary to provide a solution to the new question of the control on the Product Model created, on the fabrication and assembly documents obtained from it and what is more important, on the modifications. The traditional approach of the implementation of a standard PDM-PLM solution has been nowhere near successful enough, due to the particularities of these systems, with their time-consuming and difficult process for implementation and start-up which have annulled all theoretical advantages. This is why it is worth considering other solutions that can provide an acceptable result without exorbitant costs. The subject of this paper is the analysis of one of such solutions, its advantages and corresponding risk. This solution has recently been incorporated into the latest FORAN version, and is called Collaborative Engineering Management tool.

1. Introduction

Traditionally, shipbuilding oriented CAD/CAM/CAE Systems have not devoted too much attention to project data management, being focused only on the main aspects for which they were conceived: actual ship design and production. However, during the recent times the hard competition in the shipbuilding industry has persuaded shipbuilders that the reduction of costs and delivery times (the only way to survive) requires not only the use of advanced CAD/CAM/CAE systems, but also a better management of the project. And this better management can be achieved, among other methods, with the use of a dedicated tool that can control the Product Model and the corresponding design and production documentation..

In the field of Mechanical Engineering, the use of management tools is not new, but in the shipbuilding industry this question is comparatively new, and the existing shipbuilding oriented CAD/CAM/CAE systems requested to provide a solution to this question have been forced to link their systems with existing standard PDM-PLM systems available on the market, instead of developing their own tool. However, up to now the tested solutions have been demonstrated as non-satisfactory, in part due to the difficultness of the implementation and start-up processes of these standard PDM-PLM systems, and in part due to the particularities of shipbuilding.

As a results of the characteristics of the final product – the ship –, the amount of information to be controlled in a project is enormous. The documents to be handled are many and diverse (drawings, BOM's, NC files, calculations, permissions, certificates,...) and most of the current standard PDM-PLM systems have not been conceived for this task. However, this control is more necessary if we take into account that during the design process and even during the construction stage very often changes appear that produce the cancellation or substitution of existing documents and the creation of new ones.

On top of that, the appropriate management of the project requires not only the control of the documentation, but also of the Product Model itself as the main source for fabrication and assembly documentation, obliging to link the CAD/CAM/CAE System and the PDM- PLM System at this very level.

FORAN System, as one of the leading shipbuilding oriented CAD/CAM/CAE systems, has not remained apart from this requirement of the shipbuilding industry, and has incorporated a new module intended to solve the problems described in the simplest and most efficient way. This new module is called FTEAM, Collaborative Engineering Management tool.

2. Project Management tools in shipbuilding

2.1. Convenience of its use

The main goal of these tools in shipbuilding is to control the Product Model and the relevant documentation of the project, linking both concepts and allowing the user to automatically identify the documents related to one particular element of the Product Model, and vice versa. With this, the actual advance of the project can be controlled, and modifications intended to be included at any stage and due to any reason can be analysed, eventually approved or rejected, and checked.

If the Project Management tool can be seen to be useful when applied to a single project, then its benefits are multiplied when applied to several projects that are simultaneously under construction in the same shipyard, as it serves as a single source for controlling all the projects at the same time. Particularly interesting is the case of constructions of sister ships, as the Project Management tool can efficiently control local modifications in some units of one series.

The use of Project Management tool is specially important in military shipbuilding, in which the control of changes takes a first priority, and in which the long periods of design and construction obliges the inclusion of modifications in the different units of the series (specially weapons and detection systems).

Another important factor that should not be forgotten operative life of the ship starts at the moment when the construction finishes. As the ship is a product with a very long life-cycle (more than twenty years), it is desirable from all points of view to control it. The Product Model structure created and its link with the project documentation is the best basis to manage the life-cycle of the ship. In such a way, repair or update works are made easier and faster as control is maintained over the documents affected by any intended operation. As in the above paragraph, this task is also very important in the case of navy ships.

2.2. Problems of its use

The use of Project Management tools in shipbuilding is in its first steps, and there are only a few shipyards around the world that have advanced beyond some preliminary testing installations. Therefore, many aspects can not yet be solved, and many questions are still without the proper answer. Nevertheless, some problems are already visible and others are foreseen.

The first problem is the large amount of information handled in shipbuilding. If this situation is one of the reasons for the use of Project Management tools, then it is also a potential problem for it, as the standard PDM-PLM systems have not been conceived for such an amount of information and documentation. In this sense, it is compulsory that the system should handle a file vault to store all the information, and at the same time only its correct organisation can guarantee the viability of the system (admissible response time, automatic physical location of the files,...).

Ship design and shipbuilding processes are not as strict as could be desired, and many solutions being decided as they go along just in production. And it is clear that the use of a system foreseen to centralise and control changes and modifications would hinder and even prevent this way of working which that in some cases can even speed up the building process. Although in fact this is not a problem in the strict sense of the word, it should be admitted that the use of Project Management tools will make the building process more rigid.

Another important aspect is that the ship design process takes more time than in other engineering fields, and that during this time the ship Product Model is continually being modified although it is at least being completed. This means that the transference of the Product Model from the CAD/CAM/CAE system to the Project Management tool could not be done once, but should be done periodically and having the possibility to decide the information to maintain existing data or to

overwrite it. Linked with this problem we should realise that the structures of the information in the CAD/CAM/CAE system and in the Project Management system are different. For instance CAD/CAM/CAE systems include a “design structure” (systems, zones, calculations,...) while in the “building structure” are included the actual construction elements (spools, brackets, interim products,...). So when transferring the data it should be clear what is the actual information to be included in the Project Management system.

And finally, but not the least important problem, is the customisation of the Product Management tool. According to the suppliers of these Product Management systems, it is necessary to make a very big effort to customise the system to the particular customs of the shipyard (definition of workflows, structure of the information, approval processes, organisation charts, roles, change orders,...) before the system can be efficiently used. And this effort requires a lot of time and technical assistance from the supplier of the system that can even mortgage the viability of the implementation. The axiom “the more complex the system, the harder the implementation” is valid in this case.

The solving of all the problems is difficult, and not all the aspects can be efficiently settled. That is why the user should be aware about the difficulties of the Project Management tool, and select the System that provides an average positive answer that can justify its implementation. Those aspects that can not be satisfactorily covered should be put aside.

3. FTEAM - Collaborative Engineering Management tool

FTEAM has been conceived as a Project Management tool, called Collaborative Engineering Management Tool to facilitate both the control of the product data configuration and its documentation. The solution proposed tries to minimize the implementation and customization effort, allowing the shipyard to efficiently start with the use of the system in a very short time.

Even though FTEAM can be applied as standalone PDM system to control either the product data and/or its documentation, the full benefit and power is achieved when applied to the management of the engineering data embedded in the FORAN 3D Product Model, with which it is fully integrated. All documents generated with FORAN System can be transferred to the FTEAM environment where they can be controlled.

Like any other Project Management tool, FTEAM includes two different work environments: Documentation Management and Product Data Management. Both are linked, and while the Product Data Management implies the Documentation Management, this latter can be used as stand-alone. Moreover, the Documentation Management is commonly understood as the first step towards Product Data Management.

3.1. Documentation Management

FTEAM allows the control of all the documents of a particular project and the corresponding files, independently of their format, of their purpose and of the software with which they were generated. This control is called Documentation Management. As in any other Project Management system, inside FTEAM, document is the logic entity that represents the file intended to be controlled and managed.

Documents in FTEAM are classified by types. The definition of these types is made by the user following his own criteria, with the possibility to import previously defined types from a master project or from another specific project.

Each type of document has attached a number of attributes. These attributes can be general (revision, format,...) applied to all type of documents, or particular ones, applied only to a certain type. All attributes are user configurable, and their values can be obtained from previously defined tables, or directly input at the moment of the definition of the document.

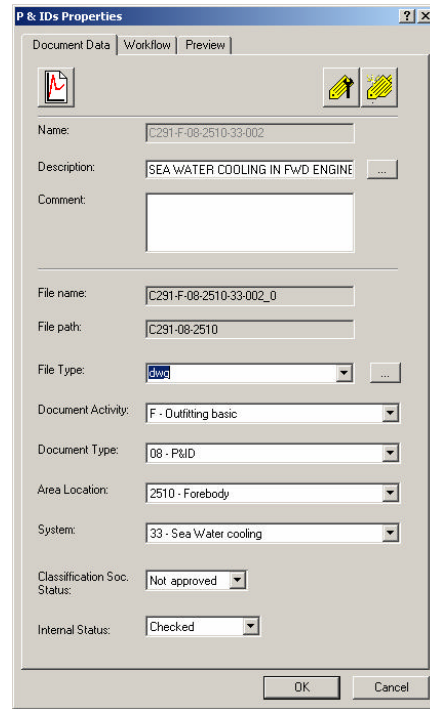


Fig.1: Example of document properties window

The organization of the documents inside the database is decided by the user, who creates a hierarchical Document Structure.

The check-in procedure of a new document in FTEAM starts with the selection of the type and the assignment to a level of the Document Structure. Afterwards, the file to be checked-in is selected, and the attributes assigned. The values of some attributes can be automatically assigned by the System, while others require direct definition from the user. This check-in procedure is more straight forward if the file has been generated with FDESIGN; in this case it is only necessary to assign the document to the Document Structure, while the definition of the document type is made automatically by the System

The documents, together with all the attributes, are stored in a relational database, while the files are physically stored in a vault. The vault characteristics are defined and modified exclusively by the administrator, being transparent for the rest of the users.

The access to a particular document requires its checking-out. The checking-out operation returns one copy of the document to the working directory of the user and locks the access of other users to it until it is checked-in again. In any case, there is always available the possibility of opening a checked-out document for visualization purposes only.

The possible operations to be made on a particular document are doubly controlled by FTEAM considering the user intending to perform the operation and the lifecycle of the particular document. From the point of the view of the user, FTEAM requires the definition of an Organization Chart of users and the authorization to carry out available tasks of each particular member of this Organization Chart. The Organization Chart includes the definition of users and groups of users in a hierarchical way, and the assignment of roles to them. The different tasks can be customized taking into account the Document Structures defined, the organization chart and the roles of each user, in such a way that each member of the organization can only perform the allowed actions. Of course the assignment of roles and authorizations is performed by the administrator of the system, guaranteeing therefore the integrity of the data.

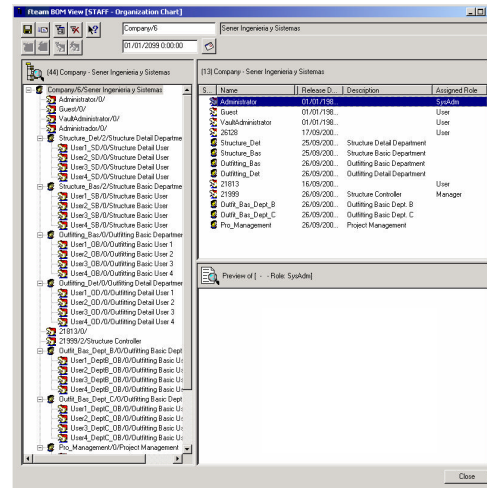


Fig.2: Example of organization chart window

The lifecycle of the documents is controlled by the corresponding workflow. A workflow basically consists of the assignment of different possible status to the documents, and the connection of these status by means of transactions. These transactions are authorized to certain users or group of users of the Organization Chart, depending on the role they have been assigned. In such a way, that actions to be carried out by normal users, managers or any other category can be segregated. Customized workflows can be defined in a graphic environment for each type of document. Additional actions can be attached to each particular status or transaction of a workflow so when the status is achieved or the transaction is carried out, the attached action is automatically started. In this way it is possible, for instance, to define procedures of automatic messaging for internal (inside the company) and external (to other companies) notification of documents changing their status.

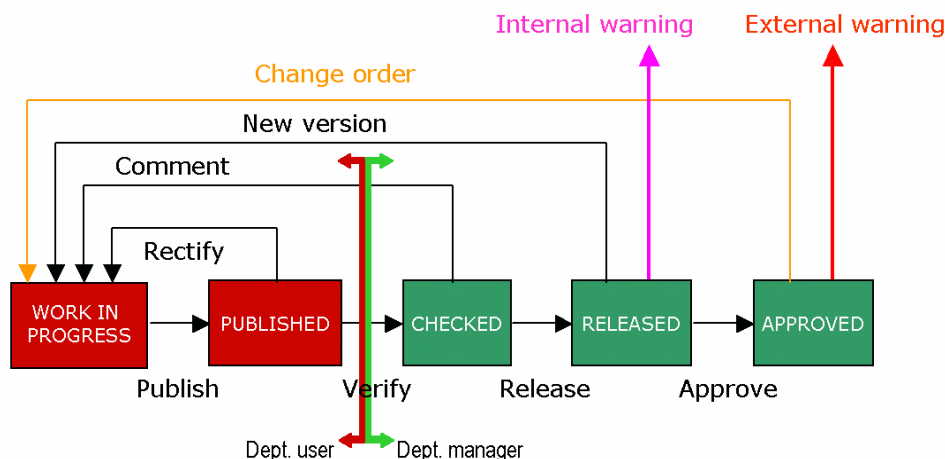


Fig.3: Typical workflow

The creation of new versions of documents is made automatically by the System taking into account the corresponding workflow. All versions of a document and the corresponding files are stored in the database, therefore it is possible to check the complete history of a document. The user has immediate access only to the last version of the document, although it is also possible to visualise old versions upon user request.

Documents inside the Document Structure can be cross-referenced. With this, the user can immediately know what documents are affected by a change.

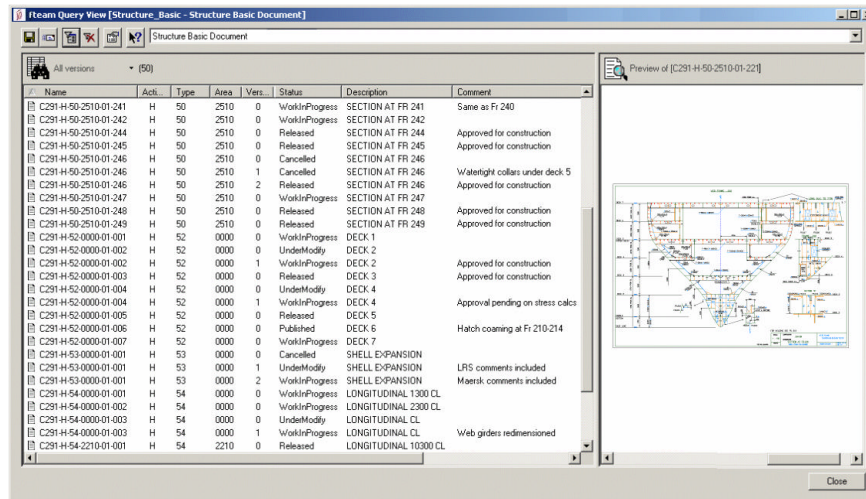


Fig.4: Example of query view on documents window

A complete set of query views on documents is available so that users can obtain information in a fully customised format. Advanced filter tools are available, allowing the user to obtain just the required information according to any concept included in the format. On top of that, the user can define additional query views for particular types of documents.

FTEAM includes a search engine that allows the display of the information required with customizable criteria. The query views allow the preview of the corresponding document, checking of its attributes, its references and the possibility of opening the attached file for viewing purposes directly from FTEAM.

Searching for information is also possible using a standard Internet connection with the help of a web browser. This allows to check the status of documents and to visualize them from remote located sites, facilitating the subcontracting of design works.

Modifications in the design can be managed by FTEAM through the concept of Change Request, which is converted to a Change Order once it is approved by the person responsible. The Change Order is then sent to the designers, including the corresponding instructions and all the documents affected by the modification. Immediately upon effecting the Change Order, versions and status of all documents involved are automatically updated according to the corresponding workflow, and different actions automatically allowed to particular users of the Organization Chart.

3.2. Product Data Management

More complex than the simple handling and management of the documentation of the Project, FTEAM allows the control of the parts of the 3D Product Model of a particular project. This control is called Product Data Management.

Inside FTEAM, part is the logic entity that represents one element (either single or interim product) of the 3D Product Model.

The handling of the Product Model in FTEAM requires its transference from the CAD/CAM/CAE. Taking into account the objective of managing the Product Model not only in the design stage, but also during the building stage and even the operation stage (lifecycle management), it is considered that, from the different Product Data Structures handled in FORAN, the best one to be transferred to FTEAM is the build strategy defined in FBUILDS. This Product Structure is in fact a structure of real elements with a physical representation that is necessary for the building process and to which in the end documents are referred.

The Product Data Management is completed by relating documents of the Document Structure to parts of the Product Structure. With this objective, once the Product Structure has been defined, documents are included in the Document Structure and assigned to one or several parts of the Product Structure.

This assignment is made automatically if the documents are generated with FDESIGN, and there are procedures to allow the global assignment of the same document to different elements of the Product Structure in a single operation. For documents not generated in FDESIGN, the procedure requires user interaction, but more automatic tools are currently under development

Any modification of parts of the 3D Product Model after their release can be controlled and authorised from FTEAM by means of the Change Request and the Change Order, that are also available for parts in the same way as for documents. If as a result of a Change Order the 3D Product Model is modified, FTEAM automatically creates a new version of the corresponding part. Documents previously assigned to the original version of the part are also transferred to the new version, warning about the necessity to update them.

A complete set of query views on parts is available so the users can obtain detailed information in a fully customised format. On top of that, the user can define additional query views for particular types of parts. FTEAM includes a search engine that allows the display of the information required filtering by customisable criteria (attributes, status, dates, size,...). The query views allow the user to navigate throughout the complete Product Structure, obtaining the information regarding the parts linked to a particular one and the documents attached to it. A preview of the parts (3D model) and documents is also available, together with the possibility of opening the files for viewing purposes directly from FTEAM.

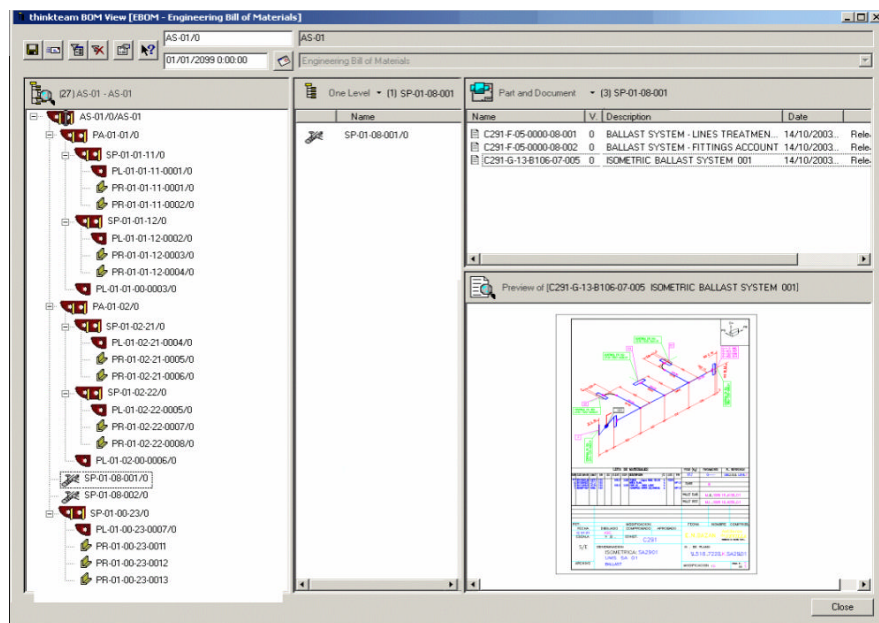


Fig.7: Example (2) of product model query window

In such a way, the navigation throughout the Product Structure allows the user to detect the parts and documents that can be affected upstream in the building process by a change in a particular part.

In the same way as for documents, searching for information for parts is also possible using a standard Internet connection with the help of a web browser. This allows to check the status of documents and to visualize them from remote located sites, facilitating the subcontracting of design works and the control of the subcontractors.

4. Conclusions

From the actual needs for controlling the Project, dedicated shipbuilding CAD/CAM/CAE Systems can not continue ignoring the necessity for providing solutions to this problem. This not being the main purpose of these Systems, the commonly accepted decision is to integrate an existing PDM System. On the other hand, the existing PDM systems are not conceived for shipbuilding, so in the integration process a high degree of customization to shipbuilding standard is required. Nevertheless, this customization requires a lot of effort, so this effort could prevent the successful use of the PDM solution.

Taking this into account, and after analyzing the pros and contras of these systems, the easiest and wisest decision is to use a Project Management tool in which the ratio effort / profit is very small.

Following this axiom, FORAN System has included in its scope a new module called FTEAM – Collaborative Engineering Management tool, which provides a suitable answer to the main problems of Project Management with a relatively reduced implementation and customization effort. In the preceding paragraphs we have tried to set out this module and its main possibilities, explaining the solution that in the author consideration, completely fits the requirements of the shipbuilding industry.

References

ABAL, D.; ALONSO, F. (2003), *Collaborative engineering in shipbuilding environment*, World Maritime Technology Conf., San Francisco

FOLLESDAL, A.; GARCIA, L. (2002), *Internet based collaboration – an opportunity to increase efficiency in shipbuilding ?*, Int. Conf. in Computer Applications for Shipbuilding, ICCAS, Malmö

ABERDEEN GROUP INC. (2003), *Data visualization – Foundation for PLM success*, Boston

HEWLETT PACKARD (2002), *PDM in brief*, Seoul

Applications of Self-Organizing Systems in Maritime Environments

Martijn Neef, TNO Physics and Electronics Laboratory,
The Hague/The Netherlands, neef@fel.tno.nl
Anthonie van Lieburg, TNO Physics and Electronics Laboratory,
The Hague/The Netherlands, lieburg@fel.tno.nl

Abstract

Maritime platforms are becoming increasingly complex to maintain and operate, due to rising demands in terms of operational capabilities, functional robustness and cost efficiency. Moreover, with an ongoing demand for reduced manning the need for more capable and clever control systems is obvious. Conventional automation approaches typically fall short on maintainability, robustness and scalability, and becoming inadequate to capture the complexity of modern maritime environments, with their abundance of data and dynamic nature. To resolve some of these concerns, the Command & Control group of TNO-FEL is researching the application of self-organizing systems theory for various types of maritime decision support systems. These biologically inspired systems are typically distributed, fast and reactive, computationally very simple, and above all, possess autonomous self-organizing properties that we can employ to build new platform control systems. The most typical example of self-organization are the so-called 'ant-based algorithms', which are a class of algorithms that mimic cooperative behavior of real ant behavior to achieve complex computations. The structural simplicity of these systems, together with their self-organizing capabilities make them a very interesting approach for certain types of computational challenges. We will demonstrate their use in damage control systems, platform management systems and decision support systems, and will describe how we envision their use in future control systems.

1. Introduction

As is the case with many domains, the maritime domain is becoming very complex and hard to deal with from an automation point of view. Maritime platforms are becoming increasingly complex to maintain and operate, due to rising demands in terms of operational capabilities, functional robustness and cost efficiency. Moreover, with an ongoing demand for reduced manning the need for more capable and clever automation is obvious. Conventional automation approaches typically fall short on maintainability, robustness and scalability, and are becoming inadequate to capture the complexity of modern maritime environments with their abundance of data, their dynamic nature and the complex inter-system relationships that characterize modern maritime platforms. To resolve some of these concerns, the Command & Control group of TNO-FEL is researching the application of self-organizing systems theory for various types of maritime support systems. This paper shortly introduces the basic principles of self-organization, and describes how they can be applied in practice. To this end, we introduce two case studies in which we utilized the concept of self-organization and elaborate on the general benefits of the approach.

TNO Physics and Electronics Laboratory (TNO-FEL) has a longstanding relationship with the Royal Netherlands Navy, and may be considered as being one of its chief research and development centers. TNO-FEL has contributed to numerous innovations on sensor technology, platform system control and weaponry, and also provides advisory services on Command, Control, Communication and Information (C3I) issues. TNO-FEL also has strong ties with commercial maritime industries, with whom in close cooperation innovative solutions are being developed in the field of decision support systems and distributed control.

2. Self-organizing systems

This section gives a short introduction to self-organizing systems and their applicability as computational optimization algorithms.

2.1 Introduction

Self-organization is a very much used and misinterpreted term in computer science nowadays. Researchers and software developers are eager to label their products as 'self-organizing' because it adds a certain mystical flavor to the functionality of a system: that of being able to 'magically' configure and manage itself without any human intervention. In reality, engineers usually refer to the capability of a system to manage itself under fairly known circumstances, such as the capability to detect certain anomalies and apply behaviors to correct these failures. This is not self-organization, but rather smart programming. To be precise, self-organization is not a technology, but primarily a way of thinking about dynamic, adaptive systems, which applies to biological, social, chemical and physical systems. Self-organization is about system structure appearing without explicit steering from outside of the system. It is about emergent global system behavior that stems from local interactions between the different entities that form the whole system, without explicit representation of these global patterns on the level of the individual components. The main characteristic of all these systems is their ability to achieve complex collective tasks and (organizational) structures with relatively simple individual behaviors, without the need for central control or hierarchy, without central models or internal representations of the environments are situated in.

An often used example to illustrate self-organizing systems is the behavior of ant-colonies, *Dorigo and Stützle (1996)*. Although single ants presumably lack knowledge of their social organization, they jointly display extraordinary complex behavior, such as their food finding behavior. They achieve their global food searching behavior through local interactions, by means of deploying *pheromone* trails. Ants release pheromone, an aromatic substance, on their way to food. Initially ants have no idea of where food is in the environment, so they wander randomly, but meanwhile leave a pheromone trail. If the ant finds food, it will wander back over its own trail, and again release pheromone. Consequently, the trail will increase in intensity. However, pheromone will evaporate over time. So, the longer the path to food, the weaker the trail will be. Ants are sensitive to pheromone, and will follow existing pheromone trails if they come across them, with a tendency to follow the strongest trail. So, if an ant finds a good, nearby food source, his trail will be picked up by other ants (since it has still has a high pheromone intensity because of its proximity), who will also release pheromone on this trail towards the food. As more ants use a particular trail, the pheromone concentration on it increases hence attracting more and more ants. Conversely, less preferable paths will not be replenished and will evaporate. This process is executed in parallel by the population of the ant colony, resulting in a swift and efficient exploration of the surroundings of the colony.

The ant colony behavior is a very illustrative example of self-organization: the appearance of structure or pattern without an external agent imposing it, *Heyligen (2001)*. There is no central controller telling the ants where to go, or what to do. Moreover, none of the ants have a notion of the state of other ants, and their interaction is indirect, by means of the environment. The ant colony has an innate capability to organize *itself*, apparently without being capable of maintaining any internal state, or any other cognitive capabilities. Ant colonies exhibit a form of *swarm intelligence*. Swarm intelligence is the general term which is used to describe the form of intelligent behavior that natural systems exhibit. Swarm intelligence takes advantage of the synergistic effects caused by the interactions between the actors (i.e. the elements of which a systems is composed), their environment, and the traces the actors leave in their environment. The environment essentially serves as the memory for the systems, in combination with the physical characteristics of the actors. This characteristic does not deny such systems any ability to learn, but it characterizes the way these system achieve global goals, i.e. by efficiently using the environment they are situated in. Many more example of swarm intelligence exists in nature, such as the building of complex structures by swarm of insects (like bridges, nests, and chains) and display of many complex tasks (like defense, foraging, social care-taking, labor allocation and so on).

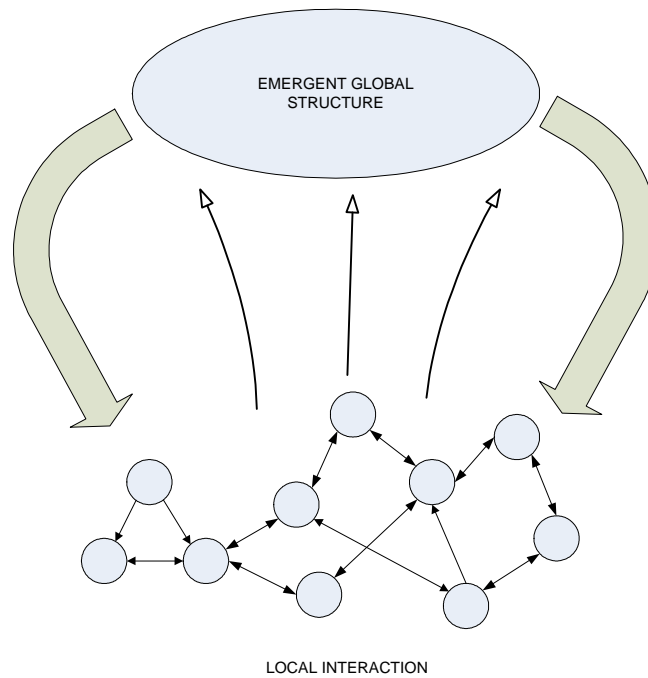


Fig.1: Emergent structure coming from local interactions (after Langton in *Lewin (1992)*)

In a broader context, self-organization is a fundamental principle of most natural systems. The main scientific theory related to self-organization is *complexity theory*, which states that 'critically interacting components self-organize to form potentially evolving structures exhibiting a hierarchy of emergent system properties', *Lucas (2003)*. In other words, complex systems (i.e. systems that are made of several closely connected parts) can, under certain circumstances, spontaneously evolve from one state to another. In terms of non-linear dynamics: any system that moves to a fixed structure (i.e. a stable pattern) can be said to be drawn to an attractor. A complex system can have many attractors and these can alter with changes to the system interconnections (mutations) or parameters. In effect, researching self-organization is equivalent to investigating the attractors of a system, their form and dynamics. Changes to a system can instigate self-organizing behavior by allowing the exploration of new state space positions. In short: if we perturb a complex self-organizing system (by for instance changing its structure or its surroundings) then the system as a whole will tend to find new stable states, and move away from unstable states along certain trajectories. Although complex systems theory is too extensive and outside the scope of this paper to describe further in this paper, it is important to note that the self-organizing properties of the applications we will describe further on have their theoretic roots in complexity theory, and in order to fully understand their potential knowledge of this line of research is essential. *Bak et al. (1988)*, *Bak and Chen (1991)*, *Lewin (1992)*, *Kauffman (1993)*, *Holland (1998)* and *Johnson (2001)* provide concise overviews to the domain of self-organization and complexity theory.

2.2 Self-organization as a Computational Approach

Many new technologies, such as evolutionary algorithms, multi-agent systems and neural networks have their roots in theoretical biology, and self-organization is no exception. It might not come as a surprise that many have quickly found their way into the domain of computer science. Their ability to tackle complex problems with only very basic rules has inspired many researchers. The use of biologically inspired computational models is twofold: they may help us in understanding biological phenomena, and they may enable new computational approaches to tackle complex problems. One of the main reasons why biological models of self-organization have attracted so much interest from the engineering community is their inherent robustness and adaptive capabilities. Anyone can confirm the robustness and adaptive nature of insect colonies: the disruption of a part of colony will most likely not be enough to get rid of them. Put an object in the middle of an ant trail, and not before long the

ants will travel around that object, and continue on their way. These are capabilities that are very much desirable in engineered systems, and hard to obtain using conventional model-based approaches. Moreover, coordination and control have always been complex issues to resolve in distributed systems, and as the world is seeing more and more networked applications and environments, these issues are gradually becoming truly obtrusive.

Parunak and Brueckner (2004) hands us five domain characteristics which might indicate when self-organizing behavior (or rather swarm intelligence) might be appropriate for engineering purposes: *discreteness*, *deprivation*, *distribution*, *decentralization*, and *dynamism*. Thus a swarming approach may yield benefits in domains that are discrete (i.e. domains that consist of discrete elements), that have strict constraints on resources (such as lack or limitation of communication means or available computing power), that need computational entities to be distributed (such as in sensor networks or telecommunication networks), that are not bound to centralized control requirements (and thus may benefit from distribution of system functionality), and that are dynamic in nature (i.e. may change over time). Not surprisingly, most successful applications of self-organization have been in domains that adhere to these characteristics. The most successful real-world application of swarm intelligence has been in network routing, where the dynamic and distributed nature of the swarming approach is particularly helpful. Ant colony based routing methods have been shown to outperform conventional routing methods in various routing domains, such as telecommunication networks, *Schoonderwoerd et al. (1997)*, *Bonabeau et al. (2000)*, and route planning (see for instance ant-based optimization of the Travelling Salesman Problem by *Colormi et al. (1992)*). Other domains where self-organizational principles have been successful and actively being research include coordination mechanisms for multi-robots systems and other agent-based systems, *Kube and Zhang (1993)*, *Pagello et al. (1998)*, *Parunak and Brueckner (2001)* and many others, adaptive task allocation and resource allocation, e.g. *Bonabeau et al. (1998)*, *Cicirello and Smith (2001)*, *Eymann et al. (2003)*.

2.3 Self-organizing Control Systems

We have briefly introduced the fundamentals of self-organization and their use as a computational approach. There are many forms in which biological self-organization principles can be applied from a computational point of view. Aside from the well-known ant-optimization algorithms, there are many more example of biological self-organization models making the transition to computational sciences. For instance, neural networks and genetic algorithms are prime example of artificial systems capable of self-organization. Also, many characteristics of self-organization can be found recognized in contemporary multi-agent systems and distributed AI research. However, this paper is geared towards the use of a specific type of self-organization, and its use for a specific type of application. The stereotypical name we have adopted for our purposes is that of *self-organizing control systems (SOCS)*: support systems that have control over and have responsibility for other, often physically embodied, systems and which make use of self-organizing properties to achieve that task. SOCS systems have the following characteristics:

- A SOCS can be defined as a group of interacting components that is functioning as a whole and distinguishable from its surroundings by its behavior.
- In a SOCS the various parts are laid out as to promote a specific function. The organizational layout of a SOCS is a critical attributing factor to the functioning of the system.
- In a SOCS self-organization is the evolution of the system into an organized form *in the absence of an external supervisor*.
- SOCS are tolerant to partial failures. Both the dynamic structure and distributed functioning enable graceful degradation and fault-tolerance.
- In a SOCS the functioning of the system as a whole is not explicitly programmed though the behavior of the entities that form the systems is defined by simple rules. Global behavior emerges from local interactions of these entities.
- SOCS are dynamic in structure. System parts may be removed from the system (e.g. in case of failure) or added to the system (e.g. ad-hoc replacements) at any time. Being open and

dynamically structured, the system does not need to be explicitly configured to accommodate for system changes.

- SOCS components rely on interaction (communication), and tend to make use of the simplest form of communication that allow the system to fulfill its aims.

In our research we primarily use the *cellular automaton* paradigm. A cellular automaton is a system made up of many discrete cells, each of which may be in one of a finite number of states. A cell or automaton may change state only at fixed, regular intervals, and only in accordance with fixed rules that depend on cells own values and the values of neighbors within a certain proximity. A classic example of a cellular automaton is Conway's Game of Life, *Gardner (1970)*. The Game of Life is a set of simple rules that governs the behavior of a two-dimensional cell pattern, that have an amazing potential to generate complex and interesting new patterns depending on the initial pattern. Each generation switches cells on or off depending on the state of the cells that surround it. This indirect interaction (the status of a cell depending on the status of its neighbor) is a typical example of local interaction causing global patterns without any central coordination, thus of self-organization. The following case studies illustrate the SOCS concept in more detail.

3. Self-organizing Systems in Maritime Environments

The Command and Control group of TNO-FEL is applying the notion of self-organization in various domains such as traffic management, decision support and platform control systems. This section described two of these efforts which are situated in the maritime domain: an example of a platform control system, and an example of a damage control system. Additionally, we illustrate the use of self-organizing systems as an integral part of a control system.

3.1 Distributed Platform System Control

Major issues in ship control system development are achieving *robustness* and *flexibility*, the former implying the ability of a control system to sustain a certain amount of damage without losing operational capabilities, and the latter implying the ease with alterations in the platform system can be accommodated in the control software. In order to optimize robustness and flexibility, we have developed a fully distributed, autonomous control systems approach, which employs self-organizing properties to sustain maximum platform system performance. As a case study for our research we have used a generalized model of the chilled water system (CWS) aboard a frigate. A chilled water system provides cooling to several frigate systems by transporting chilled water throughout the ship by means of pipes. The water within the CWS itself is cooled by seawater. A typical chilled water system consists of little more than pumps, valves and pipes. Aboard modern ships these individual components are controlled remotely from a damage control center. When an incident occurs, various alarms are sent out by sensors and monitoring equipment mounted on systems and the CWS itself.

For a chilled water system the most crucial function is proper distribution of cool water amongst various instruments. To fulfill this goal it should maintain its physical integrity as well as it can. The only means available to the control system is the opening and closing of valves, which enable the rerouting of water and cooling liquids if required. When an incident occurs that causes physical damage to the system, it is paramount to act as responsively and effectively as possible to limit the damage by prompt *isolation* of the affected sections and fast consequent *rerouting* of the water flow. In contrast to the conventional approach, where human operators (semi-)manually control the status of valves in case of an emergency, we have relocated the control of the chilled water system to an autonomously functioning architectural layer that is directly interfaced to the physical platform system. This layer, simply denoted as the *reactive* layer, is in charge of actually configuring the platform system (i.e. utilizing the actuators, such as valves, pumps, etc.) and sensing the status of the platform system (i.e. reading sensor values). An important aspect of this layer is that it is fully distributed, not only functionally, but geographically as well. Each functional entity in the platform system is represented in the reactive layer by a separate agent. This collection of agents is responsible for carrying out a kind of *reflexive* behavior: an immediate corrective reaction in response to failures

or damage of the platform system. It contains logic that enables it to quickly and autonomously change the system configuration as to minimize damage.

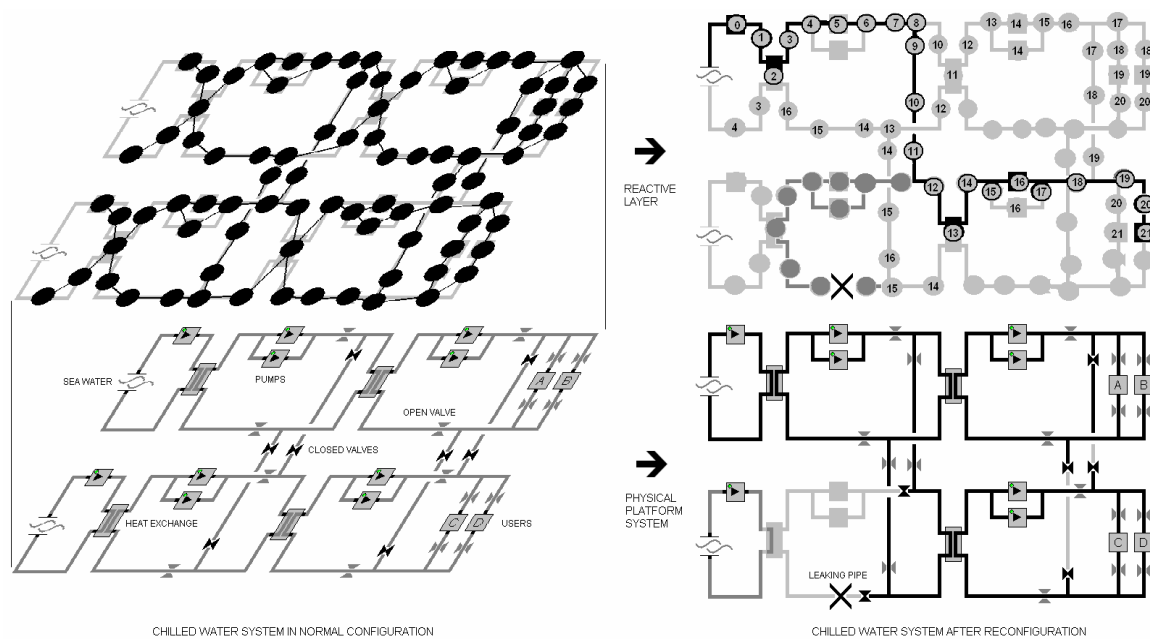


Fig.2: Illustration of the hybrid gradient routing method. On the left the agent layout with respect to the platform system, on the right the functioning of the routing algorithm.

Detecting leakage may involve knowledge of water pressure and flow in the pipes. Declining values for both values indicates that the configuration has changed, possibly because of a leak. To detect a leakage a pressure or flow meter should be placed at each few meters of pipe, covering the entire system and all of its components. Once a leakage has been detected, the first thing to do is close off the leaking part of the circuit. For this we use the following rules:

1.	WHEN PRESSURE OR FLOW FALLS SET ALARM FLAG SEND ALARM TO NEIGHBOR
2.	WHEN RECEIVED ALARM FROM NEIGHBOR SET ALARM FLAG
3.	WHEN ALARM FLAG IS SET: SHUT DOWN EQUIPMENT IF CAPABLE CLOSE VALVE IF CAPABLE IF INCAPABLE OF CLOSING PIPE, THEN SEND ALARM TO NEIGHBOR

All agents in the reactive layer behave to these rules. The agent closest to leaking pipe will detect the first fall of pressure. An alarm will be dispersed along the circuit away from the leak. The alarm stops where it meets the valves that border the circuit. These valves will be closed. The leaking circuit is now separated from the rest of the network and equipment on it has shut down.

Now that the network has changed it is necessary to reconfigure the rest of the valves to guarantee the cooling of other systems. Is it is clear that reconfiguration is an urgent task and ideally should be triggered as a reflex action. One might wonder whether the reconfiguration function can do without complete knowledge of the network. We agree on this, but we think that this knowledge should be available in fully distributed form (i.e. there is no central location where there is complete knowledge about the state of the network). To implement the reconfiguration process as a reflex action, it needs to be broken down to simple procedures for which only local knowledge of the network is required. In this form the function can be distributed over the network. We achieved this by using a hybrid distributed gradient descent method.

After a leakage has been detected, the valves that shut off the leaking circuit will send an reconfiguration request to their neighbors by using the following behavioral rules:

- | |
|---|
| <p>4. WHEN CAPABLE OF CLOSING A VALVE
AND ALARM IS SET
AND RECONFIGURATION FLAG IS NOT SET
SEND RECONFIGURATION REQUEST
TO ALL NEIGHBORS
SET RECONFIGURATION FLAG</p> <p>5. WHEN RECEIVED A RECONFIGURATION REQUEST
AND ALARM IS NOT SET
AND RECONFIGURATION FLAG IS NOT SET
SEND RECONFIGURATION REQUEST TO ALL NEIGHBORS
SET RECONFIGURATION FLAG</p> |
|---|

As a result the request is spread out over the network and all agents in the network will know that a reconfiguration will take place. To find a path from the users to the seawater we will use a hybrid distributed gradient descent method. This is implemented by the following rules:

- | |
|---|
| <p>6. WHEN CONNECTED TO SEAWATER INLET
AND RECONFIGURATION FLAG IS SET
SEND GRADIENT VALUE "0" TO OUR NEIGHBOR</p> <p>7. WHEN ALARM FLAG IS NOT SET
AND RECEIVED A GRADIENT VALUE
AND OUR GRADIENT VALUE IS HIGHER
SET OUR VALUE TO RECEIVED VALUE
SEND GRADIENT VALUE "OUR VALUE + 1" TO OUR NEIGHBOR'S UPSTREAM</p> |
|---|

Each message holds a counter. The receiver will increase this counter before it forwards the message to its other neighbors, unless it has received a copy holding a lower counter earlier on. In this way the message is distributed over all functional components in the network, which creates a 'gradient' from the seawater to the users (the devices that depend on the delivery of chilled water, such as electronic equipment). Once the gradient reaches a user, it will send back a chilled water request. This request will follow the gradient downhill towards the seawater inlet:

- | |
|--|
| <p>8. WHEN CONNECTED TO SEAWATER INLET AND RECEIVED A GRADIENT VALUE
SEND DELIVERY REQUEST TO THE NEIGHBORS HOLDING THE LOWEST GRADIENT VALUE</p> <p>9. WHEN RECEIVED A DELIVERY REQUEST
SEND DELIVERY REQUEST TO THE NEIGHBORS HOLDING THE LOWEST GRADIENT VALUE
SET "ON PATH" FLAG</p> |
|--|

The seawater inlet will receive the delivery request and confirm the request. All agents that receive the confirmation and are on a delivery path will open their valve, while all others close.

- | |
|--|
| <p>10. WHEN RECEIVED A DELIVERY CONFIRMATION AND "ON PATH" FLAG IS SET
OPEN VALVE IF CAPABLE</p> <p>11. WHEN RECEIVED A DELIVERY CONFIRMATION AND "ON PATH" FLAG IS NOT SET
CLOSE VALVE IF CAPABLE</p> |
|--|

This small set behavioral rules results in prompt reconfiguration of the CWS in such a way that, if there is an alternative route which the chilled water can follow (e.g. through cross-over pipes between chilled water zones) to reach isolated users, the network will configure itself to do so.

So, what is the use of implementing a distributed and highly decentralized system design when a centralized processing unit (like a diagnosis module) is still considered to be necessary? When a processing unit (e.g. an agent on the reactive level) cannot communicate with its neighbors, it can still handle local problems because it only needs local information to do so. What would happen when a

crewmember detects a leaking valve? That person would most likely try to fix the valve or even bypass the leak with hoses. In the latter case, the CWS will remain operational, but its architecture has changed and the affected valve is no longer part of the CWS model. This is where a distributed architecture has a clear advantage over a centralized system. The knowledge of a centralized system would need to be updated to capture the new condition, where a distributed architecture does not require a new model of the system because it never had one in the first place. Self-organizing control systems dynamically react to changes of the environment they are part of. Therefore, the approach is robust, highly scalable and adaptive.

The above approach has been demonstrated in both software and hardware demonstration, has shown that it can provide valuable support in emergency situations. Further developments include the embedding of such a self-organizing layer into a full control system setting.

3.2 Adaptive Evacuation Routing

Efficient evacuation procedures are a major issue aboard modern vessels. Especially with rising personnel and passenger capacities aboard modern ships, the increasing criticality of proper evacuation procedures is obvious, and several severe incidents over the past two decades have shown that this topic still deserves all the attention it can get. This section describes one of the efforts TNO-FEL is undertaking to increase evacuation performance: *adaptive evacuation routing*.

The main procedure of evacuation is to guide people away from an affected region into safe surroundings. The overall goal of evacuation procedures is to minimize the time it takes for all passengers and crew on board to move from the location where they were when the evacuation alarm was issued to safety (such as inside lifeboats or on a safe location on the vessel). On vessels all kinds of incidents could trigger evacuation procedures, such as fires, chemical incidents, malfunctioning systems or even impact damage on military vessels, and they can initiate at many locations around the vessel, such as engine rooms, cabins, kitchens or workplaces. Therefore, evacuation is a dynamical process: the actual progress of an evacuation is, aside from general, abstract evacuation plans, directed by the specifics of an incident, the ship geometry and ship orientation, and other contextual constraints. Consequently, optimization of the evacuation procedure is a complicated matter due to the amount of dynamic parameters involved. Fires may block main evacuation routes, or may interfere with the attack routes of damage control teams, routes may be blocked by fire, smoke or flooding or may be too small to carry the number of evacuees.

In this research we propose a fully distributed approach to evacuation routing. Our approach is in sharp contrast to most conventional approaches to evacuation routing optimization, which often use a centralized model-based approach, i.e. optimize evacuation routing using a complete model of the area. These model-based approaches cannot uphold in modern complex ships, and they are gradually becoming too complex to sustain and apply in practice. Also, we want the actual evacuation support towards the evacuees to reflect the dynamic nature of the process. We aim to create an adaptive evacuation routing system that guides evacuees to safety using dynamic signs and alarms. To achieve this behavior, we envision a network of low-cost smart devices that control sensors (e.g. smoke sensors) and actuators (e.g. direction signs or sound devices). Fig.3 illustrates the *evacuation support network* concept. In this example each section contains three types of devices: smoke detectors, direction sign and door status monitors. These devices are controlled by 'smart' network entities, that effectively represent the devices. The network layout mimics the actual corridor layout and gives us implicit knowledge about the layout of platform we are designing evacuation support systems for. We utilize this effect in our routing optimization procedure.

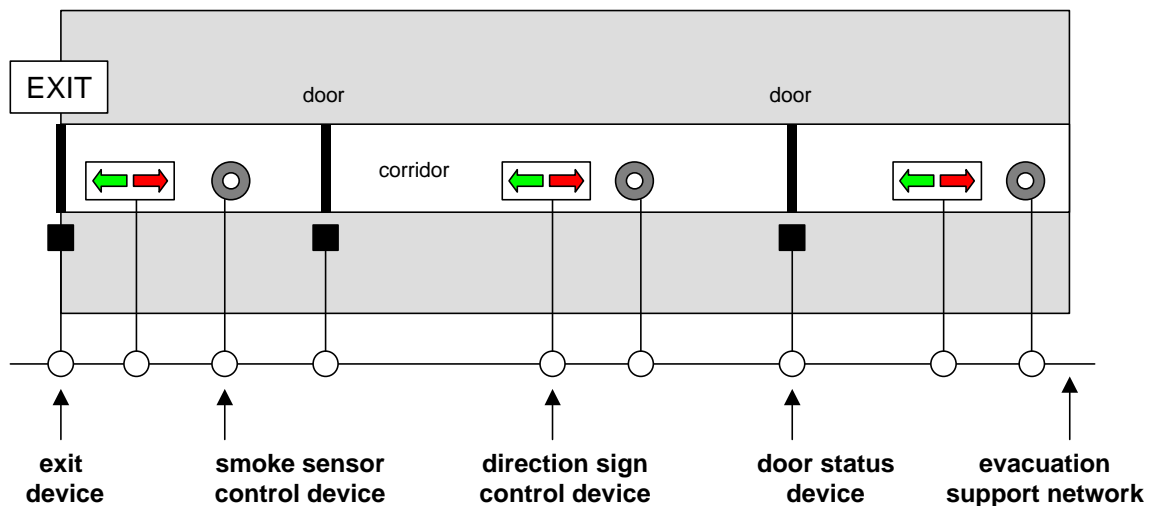


Fig.3: Illustration of the evacuation support network concept. Various sensors and actuators are controlled by networked devices to jointly provide evacuation support.

The evacuation routing process has two aims: provide optimal evacuation routes for passengers and provide efficient attack routes for damage control teams. Evacuation should have the highest priority, and should not interfere with damage control routes. The support system should inform personnel on location to where the nearest exit is, but taking into account that routes may be blocked by smoke, fire or any other intrusive circumstances. The routing procedure must also take capacity constraints for corridors and doors into account, and prevent congestion as much as possible. The nearest exit may therefore not be the most preferable exit. Our initial approach is to use a *gradient based* approach in which information propagates throughout the aforementioned network of sensors and actuators. On top of the gradient based approach we apply an ant-based optimization algorithm to deal with capacity effects of corridors and to initiate the exploration of alternative routes from any given location to an exit.

Let us examine in more detail how this would work. The evacuation process may be initiated by a alarm trigger. This might, for instance, be a smoke sensor going off. This alarm trigger subsequently sends out an evacuation initiation signal to the network. Each device in the network passes this signal on to its neighbor. As soon as exits receive this signal, they will start sending out their own signal. This signal is essentially a counter. Each device again passes on the signal, but increases the counter. In case a device already received a signal (and a counter), it only accepts the new counter if it has a lower value than the one received earlier. In effect the network becomes a kind of potential field, in which at any location the nearest exit can be found by following the gradient. This very simple algorithm is in essence enough to let direction signs point towards the closest exits, even if sections have become inaccessible due to hazardous conditions. For instance, if smoke sensors detect very high levels of smoke, their networked counterparts may block signal, and effectively prevent hazardous sections to be included in the evacuation routing. Add other types of sensors (such as camera devices or movement detection) and even more intricate constraints can be included in the evacuation routing process. As soon as the counter propagation process has finished (as may be signaled by another control signal), the direction signs light up towards the strongest signal (i.e. towards the neighboring network device with the highest counter).

The gradient-based method is good enough to handle routing, but is in this form not capable of capacity-based planning, which is essential if we deal with high numbers of evacuees. Including capacity constraints adds a dynamic aspect to routing, since we might need to disperse evacuees over multiple routes. The gradient always points towards a single exit. To this end, an ant-based optimization algorithm may be applicable in addition to the gradient based approach. We assume that it is possible to obtain information about the amount of evacuees around the vessel. This information might be obtained by using sensors, central services or even by using estimates based on general

information (e.g. time-bound information, locations of cabins and work areas, etc.). We also assume a new type of non-sensor bound network device capable of relaying information about the capacity of the corridor or section it represents. The idea of ant-based optimization (or rather ant-based exploration) is that multiple entities explore their neighborhood in parallel, and lay out gradually decaying trails while doing so, as we discussed earlier. At a certain location, perhaps a location where multiple routes to exists start, we initiate the ant-optimization process. A device sends out a number of signals onto the network, where the number of signals is relative to the amount of evacuees that need to be guided towards exists. The signals represent 'virtual ants', and the network is the neighborhood they need to explore. The network itself keeps track on how long it takes for an ant-signal to reach an exit by means of updating the signal. As they explore the network (i.e. being passed around by the network devices) they encounter the aforementioned corridor-capacity devices. These devices impose an artificial delay on the transmission of the ant-signals if too many signals arrive at once, essentially simulating a congestion. Consequently, if a signal passes through a congested corridor, its 'time to exit' increases. This information is used to make the route less attractive to newly arriving ants by lowering the gradient on that particular path, implicitly guiding the ant colony towards alternative exists. In conjunction with adaptive control procedures and signaling devices that are capable of informing evacuees about multiple routes to exists, this approach would lead to a capacity-based adaptive evacuation routing system.

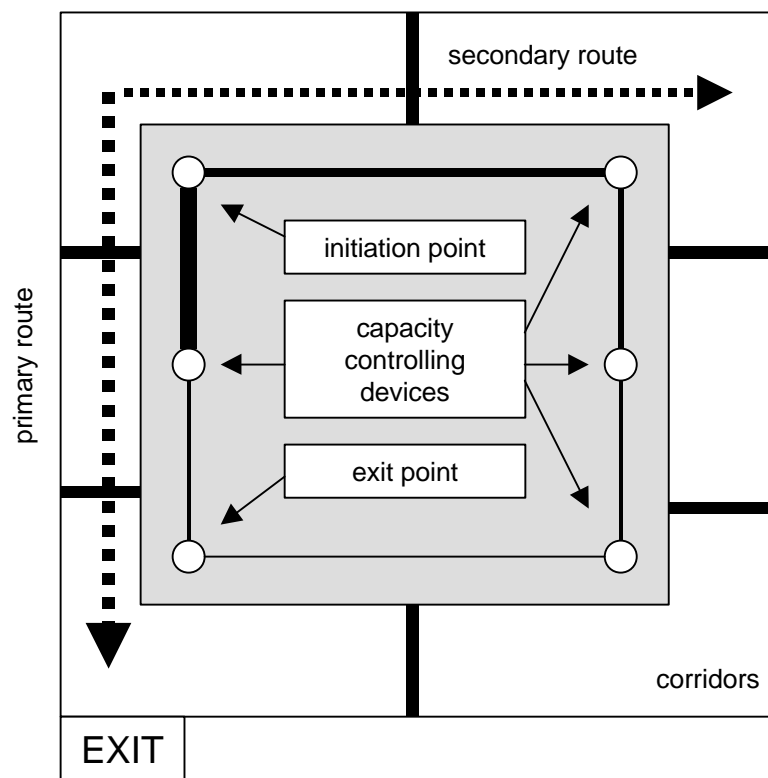


Fig.4: Network layout for a capacity-based evacuation routing system. From the initiation point 'virtual ants' explore the area, with a predisposition for timewise shorter routes. Capacity controlling devices may deliberately add a delay to the travel time of the ants, in case of crowding, leading to the activation of secondary routes

At the moment, the gradient based routing approach has successfully been simulated using an actual floor plan of a frigate. The ant-based optimization method we briefly discussed, that would enable capacity constraints, is being implemented at the moment for further examination. Nonetheless, the conceptual approach we have taken seems very worthwhile, both from a optimization point of view, as well as from a practical point of view. Even though there are very good routing optimization algorithms, we are looking for Ockam's Razor: of approaches, all other things being equal, the simpler one is to be preferred. Our approach does not require any global world-model, is robust, flexible and

adaptive. Furthermore, from a implementation and installation point of view, our approach yields even more benefits. The only requirement is that sensors and actuators that play a role in the evacuation routing process are equipped with a small (embedded) networked device. These devices are commercially available and low-cost. The procedural knowledge that these devices need to be equipped with is very simple, and in practice consist of a small set of signal-handling rules. Finally, the actual network between the devices might be tailored to the cost, quality or any criterion, as long as there is a form of communication. A very appealing line of development might be the development of an all-in-one evacuation support device, that might be composed of various sensors, a signaling device, a network devices and wireless communication capabilities to communicate with other support devices. No matter how an actual physical implementation would look like, the conceptual approach is effective and worthwhile to investigate in other domain.

3.3 Embedding Self-organizing Systems in Control Systems

Both examples in the previous section illustrate the use of self-organizing properties in operational settings. Both could be deployed as fully autonomous support systems (i.e. a fully autonomous chilled water control system, and a fully autonomous evacuation routing system). However, there is a distinct need to integrate such systems into *human*-operated control environments. No matter how much automation will be incorporated in future vessels, humans will always be responsible for the state of the vessels in the end. Especially when it comes to operations that affect the inhabitants of the vessel, human operators should be able to monitor the performance of the control systems and, if necessary, should be able to take over. However, there is a catch here. As the number of components in a platform systems is rising and the overall complexity of managing a platform system is increasing, control systems are becoming harder to design and operate. Even though command centers are equipped with state of the art control systems, trained personnel and well defined control procedures, maintaining proper control during large scale incidents still is a difficult task. Operators nowadays are faced with a multitude of information about the platform system and being hard-pressed for quick system recovery in times of critical incidents. Additionally, future command centers will also need to function with less personnel than currently, due to the ongoing demand for reduced manning. So, there is a definite need for novel approaches to platform system control in general, and the self-organizing properties we discussed earlier may play a part in achieving that ambition. This section describes how self-organizing system may be integrated into conventional control system architectures.

During normal conditions operators can deal very well with failures of today's platform systems. Small failures can be handled accurately and solved without too many problems. The real challenging situations occur during extensive incidents. In these situations operators might be overburdened, occupied with other tasks or unavailable at all, which can lead to costly loss of time before the actual damage control procedures are initiated. Quick response to an emergency is essential. In our vision, future control systems should serve as the first line of defense in case of an emergency: the control system should operate on its own to remedy the situation until the operator is able to take over, even if this would imply that the control system defines a sub-optimal solution. The control system should be able to manage itself in such a way such that at least a minimal set of operational demands are met, such as quick isolation of damaged areas, the immediate initiation of evacuation procedures or promptly turning off equipment to prevent further damage. It thus implements a *reflexive* behavior that is, to some extent, comparable to those found in biological systems (like the withdrawal reflex when we burn ourselves at a hot plate). These reflexes could help to retain the system from harmful conditions and maintain its functional integrity as long as possible. More effective and elaborate recovery plans can be formulated when the operator is back in the loop and the situation has become less critical. This kind of behavior is desirable, since it adds to the overall capabilities of the operator: the control system as a cooperating actor, instead of a mere control panel. Reflexive behavior helps to recover platform system operation as quickly as possible in case of a crisis. Our aforementioned self-organizing control system concept is a concrete architecture capable of providing such reflexive behavior. However, this kind of behavior only pays off if the control system is able to exert the proper control commands on the platform system in case of an incident. In a centralized, monolithic control

system sensors and actuators obtain their control data from a central control location, from which all decisions are made and fed back to the actuators. Of course, this is not a good approach if we aim for a robust arrangement: in case of an extensive emergency communication lines may be cut, leaving the platform system unattended for. To ensure robustness of the control system itself we need to dispose of the centralized design paradigm: we need an arrangement in which the control intelligence is distributed. The components of the control system that enable the reflexive behavior need to be close to the equipment they are controlling. Furthermore, the behavior of these components should be devised in such a way that the system is adaptive. Adaptivity is necessary because in advance it is unknown which parts of the system could be damaged. The architectural concepts we introduced in the two case studies adhere to this notion: agent-based networks that are both geographically and functionally close to the physical systems they control. However, an equally important question is how a such control system should interact with and aid human operators. In most current platform control systems the amount of support from the control system to the operator is. With larger number of actors and devices in control systems it becomes harder to provide the operator with an accurate and complete picture of the current situation without overwhelming the operator. Therefore, control systems need to display some level of intelligent cooperative behavior: the control system should actively participate in the control task. By letting the control system take over elementary tasks (such as putting the platform system in a certain configuration or gathering information), the operator could spend more time on knowledge intensive tasks such as detailed diagnosis and recovery planning. In such a collaborative setup, the purpose of the control system would be to amplify strengths and alleviate weaknesses of the operator by providing more or new resources, such as time or performance, *Chalmers (1998), Terveen (1995)*.

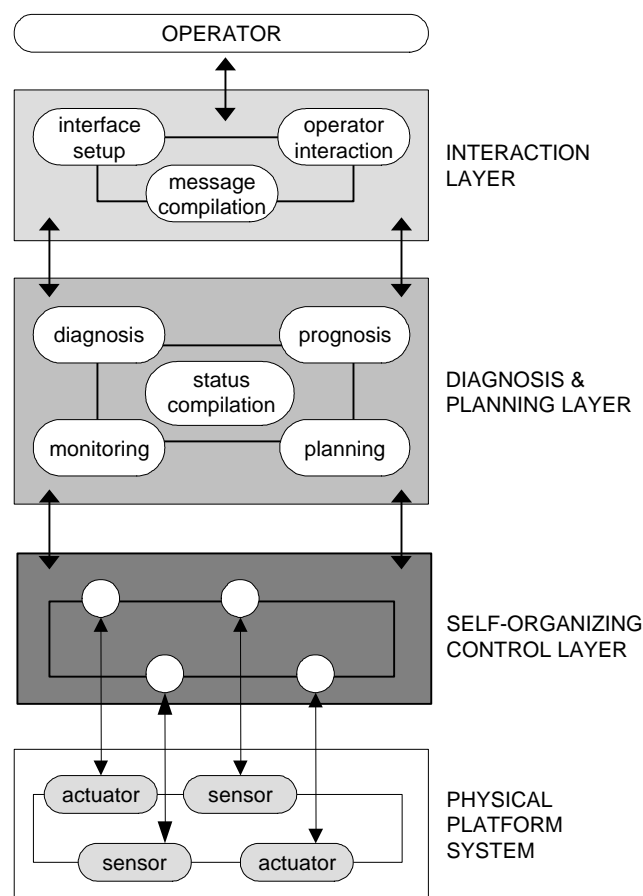


Fig.5: A layered approach to platform system control with an embedded self-organizing control layer

The main objective for our approach is threefold: the architecture must display *robustness* (be as fail-safe as possible), easily *scalable* (easy to extend when the platform system is modified) and must *facilitate* control tasks for the operator. (i.e. prevent alarm flooding, actively support the operator and provide new ways to exert control). Above all, we advocate distributed control. This implies that control tasks are performed by different parts of the control system. To arrive at a functional architecture we have defined behavioral layers for each of the functions that we want to automate and arranged them hierarchically in terms of level of abstraction. Translating data into meaning information takes time. If we're aiming for fast response, we should not be dependent on the completeness and availability of information. The less information needs to be gathered and interpreted the more responsive the action can be. Therefore the most responsive actions (i.e. the reflexive behavior) should be placed at the lowest level of abstraction. Functions that depend on more detailed information such as analysis, diagnosis and scheduling are placed at a higher level.

The proposed architecture exists of three functional layers, Fig.4: a self-organizing control layer, a diagnosis and planning layer and an interaction layer. These layers are connected in a subsumption-like manner, *Brooks (1991)*, although other schemes than priority-based selection might fit as well, in which the self-organizing control layer can function on its own and the diagnosis and planning layer can function without the interaction layer. The *diagnosis and planning layer* is responsible for diagnosis, prognosis, condition monitoring and action scheduling. This layer takes on information from the self-organizing control layer and can issue control commands back to the agents in that layer. Just like the self-organizing control layer, this layer is built up using a distributed architecture, i.e. a collection of agents. Conceptually, all functions that one would expect from a decision support system can be placed here. Depending on functional requirements of the control system one may leave out certain aspects without doing harm to the overall concept. The *interaction layer* deals with all operator interaction. It takes care of displaying relevant information in the right form, aids the operator in information gathering, provides feedback if desired, filters irrelevant data and so on. This level is also responsible for proper processing of operator commands. It could, for example, allow the operator to issue a general configuration statement without having to deal with the precise actuator settings required to achieve that particular configuration. The functional meaning of this level would be alike a smart 'personal buddy' for the operator. The information about the system that this assistant would need to have to provide support comes from the subordinate diagnosis and planning layer. Characteristic examples for the intended agent type in this layer can be found in *Maes (1994)* and *Sycara and Zeng (1996)*.

In essence, the self-organizing control system becomes a pro-active functional entity within the platform control system. It may be authorized to perform certain functions, under certain circumstances on its own, and also provides indirect access to the physical system, and thus forms a natural extensions to most conventional control systems.

4. Conclusions

The topic of this paper was the application of self-organization principles in maritime environments. We have given a short introduction to some basic concepts and have demonstrated their use in two practical case studies: the control of a chilled water system and evacuation routing. It is important to note that the approach advocated in this paper is a specific type of self-organization, and that there are many more ways in which self-organizational principles can be applied to deal with computational challenges. The most important message this paper aims to convey is the ideology of using self-organizing system models: the use of very simple structures to solve complex problems. From an engineering point of view, self-organization is not just another technology that one can simply adopt. It requires a shift in thinking about systems engineering: in order to obtain more system capabilities, we need to let systems control themselves, and make smart use of the surrounding that the systems are in (e.g. utilize their *embodiment* and *stigmergetic* capabilities). As for opportunities for maritime environments, there are many more than the two examples we have discussed. In general any type of computational problem that deals with routing or autonomous systems coordination can be approached with a self-organizing systems tactic. Especially in the field of multi-agent system design

(be they software agents or embodied agents such as autonomous robots) there are ample opportunities to employ such techniques. And with inherent adaptiveness, robustness and scalability as major benefits, this line of research should appeal a wide audience within the maritime domain.

References

BAK, P.; TANG, C.; WEISENFELD, K. (1988), *Self-organized criticality*, Physical Review A 38, pp.364-374

BAK, P.; CHEN, K. (1991), *Self-organized criticality*, Scientific American, January 1991, pp.46-53

BONABEAU, B.; HENAU, F.; GUERIN, S.; SNYERS, D.; KUNTZ, P.; THERAULAZ, G. (2000), *Routing in telecommunications networks with 'smart' ant-like agents*, (Eds. Albayrak and Garijo), Intelligent Agents for Telecommunication Applications, Springer, pp.60-71

BONABEAU, E.; SOBKOWSKI, A.; THERAULAZ, G.; DENEUBOURG, J.L. (1997), *Adaptive task allocation inspired by a model of division of labour in social insects*, Lundh et al. (Eds.), Bio-Computing and Emergent Computation, World Scientific, pp.36-45

BROOKS, R.A. (1991), *Intelligence without representation*, Artificial Intelligence 47, pp.139-159

CHALMERS, B.A. (1998), *On the design of computer-based C2 decision support for a modern frigate*, 4th Int. Command and Control Research and Technology Symp., U.S. Department of Defense, C4ISR Cooperative Program, Nasby Park

CICIRELLO, V.A.; SMITH, S.F. (2001), *Insect societies and manufacturing*, IJCAI-01 Workshop on Artificial Intelligence and Manufacturing: New AI Paradigms for Manufacturing, pp.33-38

COLORNI, A.; DORIGO, M.; MANIEZZO, V. (1992), *Distributed optimization by ant colonies*, 1st European Conf. on Artificial Life, pp.134-142

DORIGO, M.; STÜTZLE, T. (2002), *The ant colony optimization metaheuristic: Algorithms, applications and advances*, (Eds. Glover and Kochenberger), Handbook of Metaheuristics, Kluwer Academic Publ.

EYMAN, T.; REINICKE, M.; ARDAIZ, O.; ARTIGAS, P.; FREITAG, F.; NAVARRO, L. (2003), *Self-organizing resource allocation for autonomic networks*, DEXA Workshops 2003, pp. 656-660

GARDNER, M. (1970), *Mathematical games: The fantastic combinations of John Conway's new solitaire game 'Life'*, Scientific American 223, pp.120-123

HEYLIGHEN, F. (2001), *The science of self-organization and adaptivity*, The Encyclopedia of Life Support Systems, EOLSS Publishers

HOLLAND, J.H. (1998), *Emergence: from chaos to order*, Helix Books

JOHNSON, S. (2001), *Emergence: The connected lives of ants, brains, cities, and software*, Scribner

KAUFFMAN, S. (1993), *The origins of order*, Oxford University Press

KUBE, C.R.; ZHANG, H. (1993), *Collective robotics: From social insects to robots*, Adaptive Behaviour 2/2, pp. 189-219

LEWIN, R. (1992), *Complexity: Life at the edge of chaos*, Macmillan

- LUCAS, C. (2003), *Self-organizing systems*, in FAQ for Usenet group comp.theory.self-org-sys, (The Complexity & Artificial Life Research Concept for Self-Organizing Systems), <http://www.calresco.org/sos/sosfaq.htm>
- MAES, P. (1994), *Social interface agents: Acquiring competence by learning from users and other agents*, Working Notes of the AAAI Spring Symp.on Software Agents, Stanford, pp. 71-78
- MacGREGOR SMITH, J. (2001), *Evacuation networks*, Encyclopedia of Optimization, Vol. 2, C.A. Floudas and P.M. Pardalos (Eds.), Kluwer Academic Publ., pp.36-44
- PAGELLO, E.; MONTESELLO, F.; DANGELO, A.; GARELLO F.; FERRARI, C. (1999), *Cooperative behaviors in multi-robot systems through implicit communication*, Robotics and Autonomous Systems 29, pp.65-77
- PARUNAK, H.V.D.; BRUECKNER, S. (2001), *Entropy and Self-Organization in Multi-Agent Systems*, 5th Int. Conf. Autonomous Agents (Agents 2001), Montreal, ACM, pp.124-130
- PARUNAK, H.V.D.; BRUECKNER, S. (2004), *Engineering Swarming Systems*, (Eds. Bergenti et al.), Methodologies and Software Engineering for Agent Systems, Kluwer
- SCHOONDERWOERD, R.; HOLLAND, O.; BRUTEN, J.; ROTHKRANTZ, L. (1997), *Ant-based load balancing in telecommunications networks*, Adaptive Behavior 5/2, pp.169-207
- SYCARA, K.; ZENG, D. (1996), *Multi-agent integration of information gathering and decision support*, 12th European Conf. on Artificial Intelligence (ECAI '96), Budapest, pp.549-556
- TERVEEN, L. (1995), *An overview of human-computer collaboration*, Knowledge-Based Systems 8/2-3, pp.67-81

Computer Integrated Planning and Resource Management in Shipbuilding

Christian Massow, Logimatic Software, Bremen/Germany, chm@logimatic.dk
Ivan Siksne-Pedersen, Logimatic Software, Aalborg/Denmark, isp@logimatic.dk

Abstract

High complexity of product and processes, parallel processing, high customer influence are the most important characteristics of order processing in today's shipbuilding industry. This requires special, dedicated methods and tools for planning and resource management. Integration of these "best-of-class" solutions in CIM environments is the challenge to work effectively on basis of common information in all areas of the shipyard. Organization and software systems to support an integrated planning and control approach are discussed. An example illustrates a practical integrated IT solution for shipbuilding taking into account aspects of integrated planning and resource management.

1 Introduction

Due to global competition the shipbuilding industry is under pressure to improve its manufacturing efficiency. The economical situation of the shipbuilding industry is influenced by an increase in demand to improve integrated information and control concepts. The efficiency of production/project management and coordinating activities within complex production environments is expected to increase by integrating operations on various levels of planning and control. These levels include both intra-organizational operations as found in autonomous, de-centralized areas as well as operations to be coordinated between multi-site production facilities such as between manufacturer, supplier, and sub-supplier.

In this situation adoption of well-known production management concepts is not useful since the shipbuilding process, as an good example of one-of-a-kind production, shows significant differences compared to repetitive production.

As these differences, concerning customer order processing and the production process, necessitate specific requirements for a pursued concept of production management, an individual approach for the shipbuilding process is required.

2 Planning and Control in Shipbuilding

The special features of customer order processing in shipbuilding lie in the individuality of the product and the required production processes together with their corresponding dependencies. The main characteristics influencing the required production planning and control concept are:

- Due to the short lead times (relative to product and process complexity), the large-scale orders/projects require parallel production activities. This includes the simultaneity of design, procurement, operations planning, manufacturing and assembly.
- Offer planning is extremely important since every offer involves such a large outlay of financial and production resources that the existence of the organization is at risk. Moreover, the relationship between orders acquired and offers to be prepared has deteriorated over the past years and the offers must be worked up under increasing time pressure.
- Customer influence does not end with the signing the contract. In fact, the customer can permanently control product specification as far as actual product delivery. Process specification in turn is also affected of course.
- Product complexity and decreasing manufacturing penetration in terms of lean production leads to inter-organizational production. This inter-organizational production necessitates additional efforts for production planning and control.

- As a result of the above-mentioned aspects of shipbuilding, the products are produced parallel to acquisition of product and process information. The product and process data model growing slowly implicates incompleteness and inaccuracy of scheduling and capacity planning data during the whole order phase.
- In shipbuilding, human capabilities and accumulated experience of professionals are key resources. To support the possibility of improvisation and the necessary flexibility of the employees, different production areas need maximum autonomy, also in the areas of planning and control.
- Shipbuilding is a complex, complicated and multi-stage production process with high interdependencies. The process combines different manufacturing types at one site (from line manufacturing until construction site assembly).
- Due to the product variety, various temporary and capacity related limited bottleneck resources, for example dry docks and outfitting centers, have to be handled by the production planning and control center.

These characteristics imply certain requirements to the organization and supporting IT tools discussed below.

2.1 Basic Principles

2.1.1. Hierarchical Models

At the onset of planning, the information pertaining to a project is not always complete. To deal with this incompleteness of information and to provide the needed flexibility a hierarchical approach is required. This is true for all elements of the planning model, e.g. product model, resource model, process model.

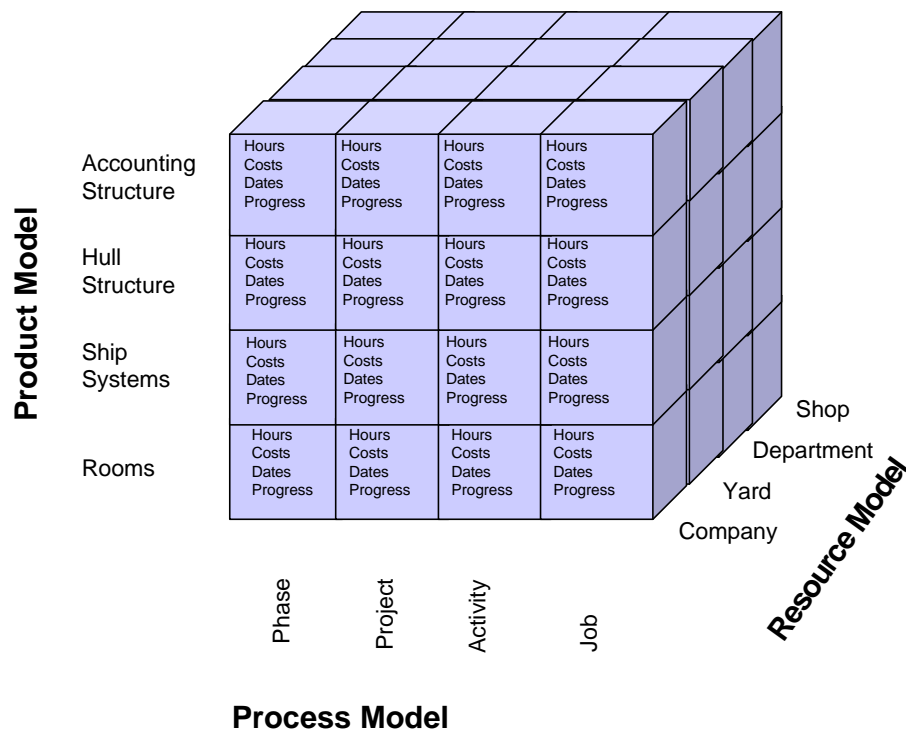


Fig.1: Interlinked hierarchical models

Elements of the process model need to be related to each other. E.g. activities should be assigned to production phases like engineering, hull fabrication, assembly, outfitting Additionally the several hierarchical models are to be linked between each other. Hence a product oriented project management is supported. The set up must be able to answer questions like:

- What is the status of section 711?
- When will room 0815 be ready for carpenters?
- What are the actual cost for the water supply system?
- How much hours were spent for outfitting of block 4711?
- What does the assembly phase costs?
- ...

2.1.2. Modularization

A module is a complex entity within a complete system which consists of several elements and is self-contained, hence compatible. The principal of modularization requires the decomposition of the product, processes, and resources into a limited set of such modules. Modules should be created in a way that they have strong bonds inwards and loose links between each other. In this way, the modularization helps to reuse the modules in other contexts. This goes for the configuration of the product as well as for the production system (processes and resources).

Reference modules e.g. support the planners creating a new planning model by providing them with standard objects. The classification of a reference module is done according to certain product attributes which lead to different process parameter. The reference modules library is a hierarchical set of modules with respect to the common product structure, Fig.2.

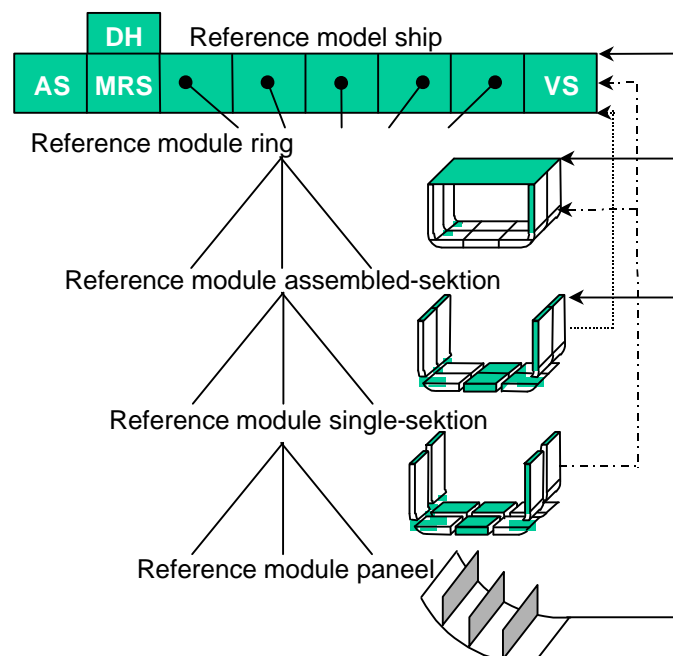


Fig.2: Reference module hierarchy

Hence the planning and control architecture must also follow an hierarchical modular approach to be able to make use of the different reference modules on different levels.

Table I: Exemplary matching of Reference Modules and Planning Elements

system	standard planning element	reference module
project management	project version; capacity demand; network; sub-network; part network; activity; type curve	ship; ring; assembled section
material management	drawing part list; work order; section 0	single section; panel
work shop management	product-activity relations; work order responsibilities	assembled section; single section
shop floor monitoring and control	jobs, picking lists, pallet list	Panel

2.1.3. Project Specific Procurement

The one-of-a-kind background of shipbuilding requires special emphasis on project specifics. Material causes 60-70 % of the costs of a new building if not more. Savings in this area will have the greatest impact on the total costs. Hence procurement is a success factor for shipyards. The most difficult procurement processes for shipyards are those related to project specific components. The process is difficult because, *Siksne-Pedersen (2000)*:

- Need for early purchasing of “key” items.
- Information required is created iteratively and final information is often only available after a purchase contract is placed.
- The purchasing process is very technically driven and requires teamwork between designers and buyers.
- Purchase of material is often made before the final necessary date in production is decided.
- Steel material (plates and profiles) often need to be treated as project specific material.

It is extremely important to focus on the key items for a project. The key items are the project specific material that is defined, purchased and used for a project. This is often the material with the longest delivery time and are the most critical items for the material handling. Buying in a wrong quantity will affect the material costs as illustrated in Fig.3, *Siksne-Pedersen (2000)*.

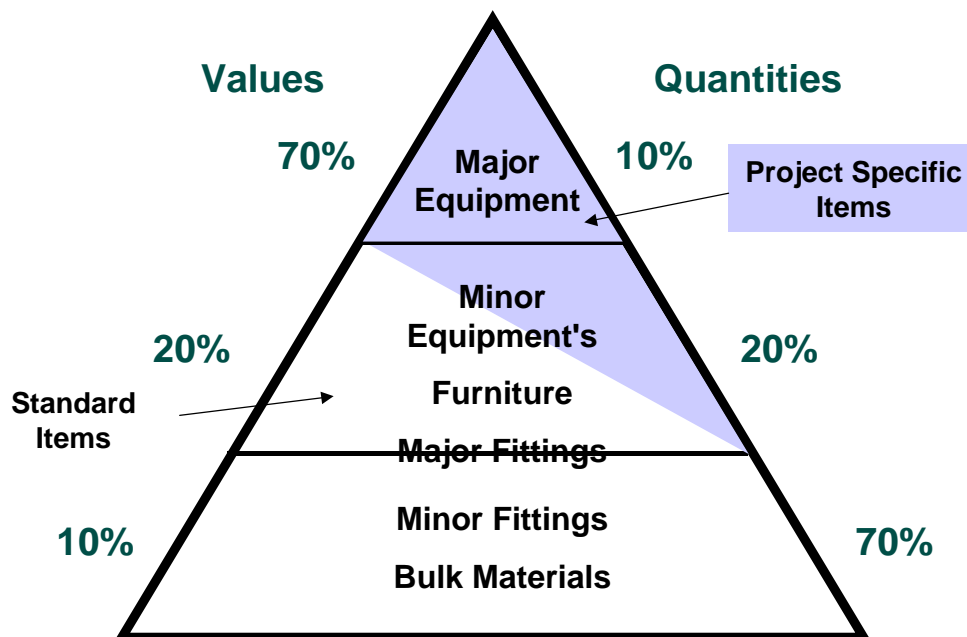


Fig.3: Material Triangle

The material triangle shows the importance of material management when building e.g. a ship. The project specific components makes up for approximately 70% of the costs in such a project, so accurate management of the material and logistics is vital. Buying at the wrong time (or for the wrong time) will affect the material costs as well. Additionally the production costs will be increased. Thus the importance of integrated planning and procurement is obvious.

2.1.4. Part listing

The part listing process in shipbuilding is split in two different areas: Part list for building of hull, and part list for outfitting. Both part list types are used to support the material control process in relation to purchase requirements and delivery of material to production.

Part list for hull: The part lists created for the hull describe a hierarchical break-down of the entire steel structure into elements such as blocks, section, sub-assemblies, and panels. This parts list supports the building of the hull. Traditionally material like steel plates and profiles has been considered stock material and purchased in bulk. For supporting the purchase process for buying steel and decrease the surplus amount, it is necessary to have a separate steel estimating function for defining and reservation of steel requirements per section/block. This estimation is then used when buying steel and when planning the delivery of plates and profiles to the different shops. This shows again the need to integrate planning, *Siksne-Pedersen (2000)*.

Part list for outfitting: The part listing process consists of building up the material needed for the different design units. A design unit can be an assembly, a drawing, a pre-fabrication or similar. By relating the material from the catalogues to the parts list the material list for the design unit are defined. It is required to create a hierarchy of different design unit by relating these to each other.

The part lists for outfitting have two main purposes:

- Defining the material requirements.
- Support of the outfitting process.

2.2 Aspect of Integration

One can distinguish between the integration of planning and the integration of production supporting processes in general. For the integration of planning, one will find a lot of different, most likely independent plans and schedules within a shipyard. Starting from the long-term yard program, an enormous number of decisions must be made and stipulated in a plan. A plan is the definition of future actions. As time proceeds the planning horizon is getting shorter and details increase, thus the number of plans increases as well. Nevertheless the different plans – or the underlying decisions – have an impact on each other. Therefore the plans must be integrated in two respects:

- Horizontally in sense of long term and short term decisions
- Vertically in sense of Rough and detailed planning and operation

The other point of view towards integration deals with the business processes. If one looks at the tasks of order processing as there are Design, Work Preparation, Procurement, Manufacturing, Assembly, Delivery ... it is essential to work on the basis of common information. With respect to planning and resource management this requires the integration of process scheduling, resource allocation and material requirement planning. The resources to be considered in shipbuilding are of quite different nature. Additionally to the classical resources man power and machinery we find some others which require special attention.

- The management of project-specific material is a challenge in shipbuilding
- Information is a critical resource for shipbuilders. Drawings, programs, test instructions etc. must be available at the right time at the right place (in the correct format).
- Some production floor space (dry dock, slipway, assembly hall, painting boxes) is expensive and rare.

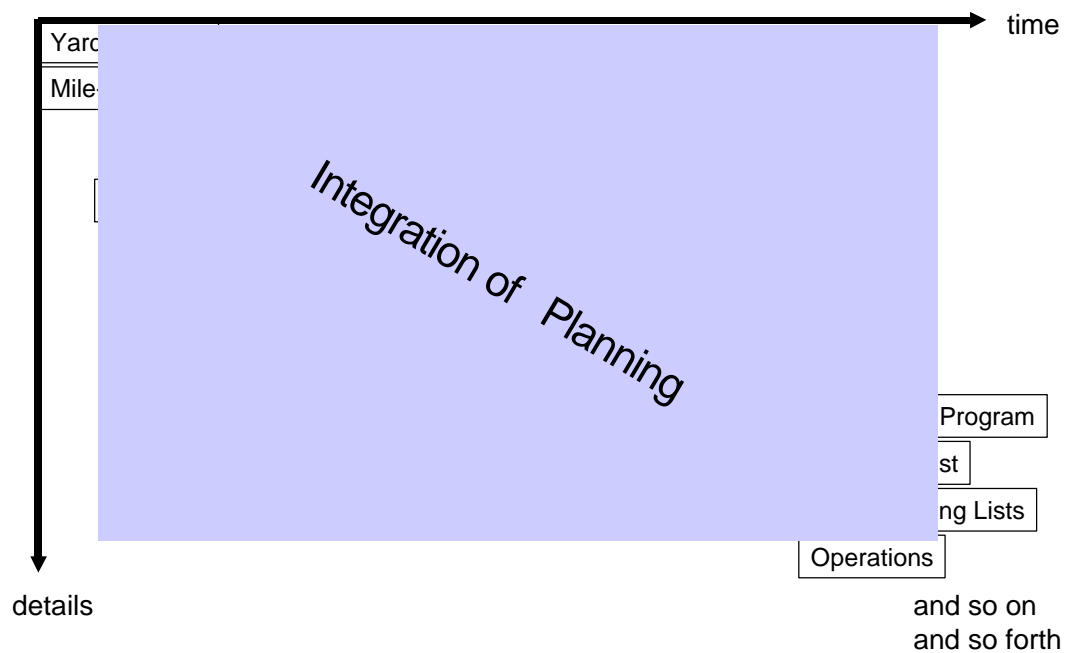


Fig.4: Integration of plans

Planning must not stop at the definition of start and end dates for production activities, but must insure the feasibility of the defined future actions. Hence the planning of the process and of its input (resources) as well as of its output (resource for successor process) must be integrated.

3 Integrating IT Solutions

IT systems at shipyards are the information backbone on which the entire organization has to rely. Consequently there is not room for functional compromises. It is vital to have transparency and to react in real time. This requires diverse dedicated and vertical IT systems comprising CAD/CAM, planning, material management and finance/HR.

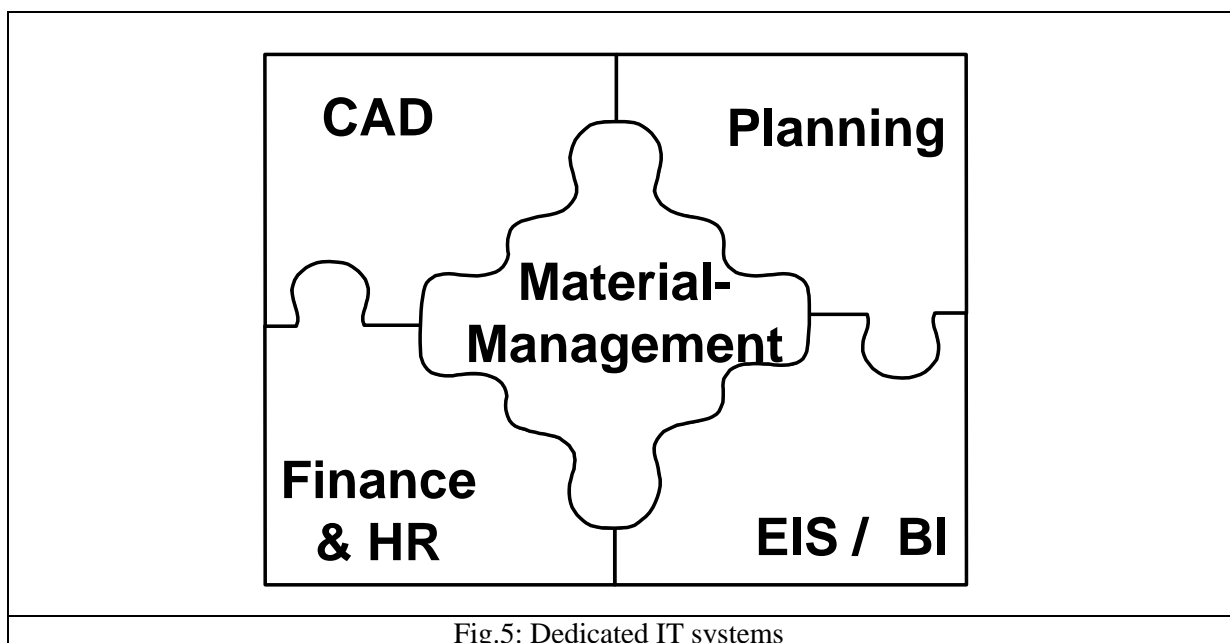


Fig.5: Dedicated IT systems

The shipyard industry changes and so do the requirements at a shipyard. The 'best-of-class' approach and strategy provides a solution which does not compromise functionality and which gives the possibility to upgrade each application as appropriate and required. For most shipyards it is very important to have the possibility to change and replace part of the IT applications without distorting the overall IT architecture.

With respect to computer integrated planning and resource management in shipbuilding we must have a closer look at the integration of dedicated solutions for planning, control and material management. We discuss techniques to integrate dedicated systems first. Then we describe the functional integration of such systems.

3.1 System Architecture

Fig.6 integrated system architecture shows a strong basis that allows IT systems to be integrated. A rectangle denotes the presentation and function tiers, the drum symbol denotes the data tier and arrows denote data exchange, *Torp et al. (2002)*.

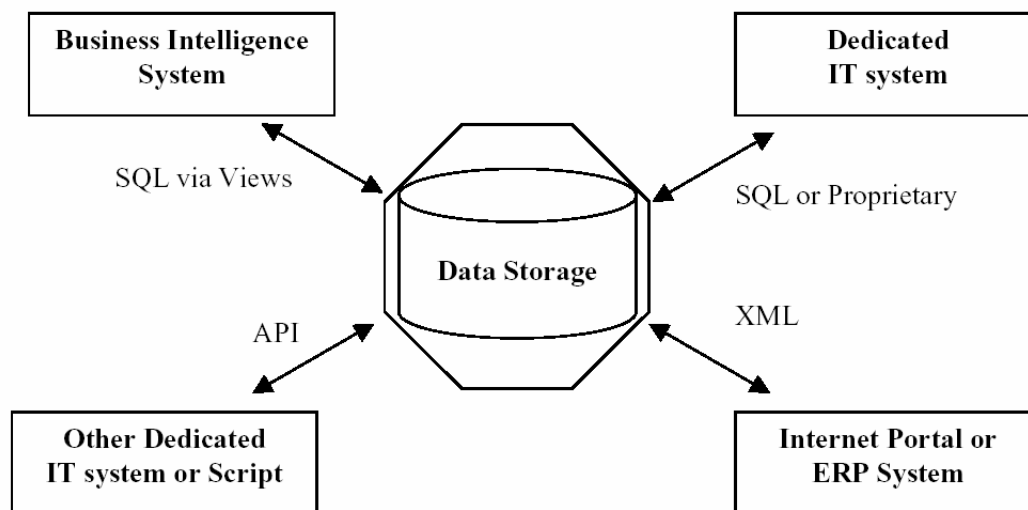


Fig.6: Integrated System Architecture, *Torp et al. (2002)*

The data storage is still provided by the supplier of the dedicated IT system and these are still closely connected. However, the data storage has been made much more accessible to other IT systems. Each IT system still provides its own data storage (not shown in Fig.6). It is not the intention that all IT systems should share the same data storage. At least in principle, all systems have equal access to all the data storage. However, we believe it will never make sense to exchange all information to all IT systems. As an example, it does not make sense to send a 3D CAD drawing to an ERP system. The data owner must be responsible for granting access to the other IT system and make sure that only correct and valid data is presented in alternative ways and formats to other IT systems, *Torp et al. (2002)*.

The octagon surrounding the data storage is a thin layer of code from the function tier encapsulating the storage. Providing direct access to the data storage would couple the IT systems too hard and not allow each system to evolve independently. Fig.6 also shows that there are several different ways to facilitate a data exchange, as indicated by the labels on the arrows. This can be done by query language SQL, via an API, or via XML documents. In the following section we discuss the advantages and disadvantages of the three approaches (XML, SQL, API) that the integrated system architecture provides for integrating IT systems.

The arrows in the figure depict several different ways that IT systems at a shipyard can be integrated.

In the following we start with a discussion of role of the dedicated IT system when the system architecture evolves.

In Fig.6 the dedicated IT system is the presentation and function tiers of a single system. The dedicated IT system is basically retained in the transition from an isolated architecture to an integrated architecture. The dedicated IT system still provides the main entry for a single system for the human users at a shipyard. Examples of such systems are a CAD system, a planning system, and a material control system, *Torp et al. (2002)*.

3.1.1. API approach

A frequently used approach is the use of an API for integrating IT systems, Fig.7. The API which is implemented in a specific programming language can be used by both the supplier of the API, i.e., the supplier of the dedicated IT system and other IT systems at the shipyard. Here System B calls System A's API. In Fig.7, we assume that this API has the three methods ('exist', 'delete', and 'get'). This list of methods could be expanded.

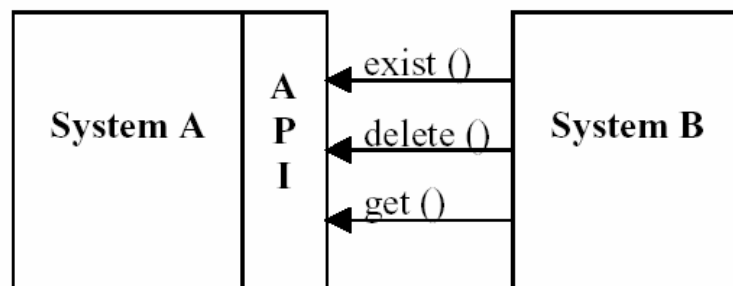


Fig.7: API approach, *Torp et al. (2002)*

One of the main benefits of an API approach is that the integration can be very close and each IT system provides its full potential in the integration. Such a close integration can be very cost-efficient for the shipyard. Since an API typically provides a high level of context (i.e., first a connection is opened, then data is queried and exchanged, and finally the connection is closed), the API approach is well suited for the exchange of many small objects between the IT systems. The API approach can be used geographically both internally and externally. This depends mostly on the programming language chosen and the underlying operating system. Either push or pull principle can be used with an API approach. This can be implemented by using the API in triggers implemented on the underlying data storage.

The major drawback of an API is that it is a vendor specific. This lack of standard requires that the shipyard must become familiar with each vendor's API or rely on the vendor to implement an integration. This requires that a vendor must be committed to supply a stability and backward compatible API to preserve the shipyard's investment. The stability requirement is typically also in the interest of the vendor and we see no conflict here. A second drawback of an API approach is that it is typically only available in a few programming languages, e.g., Java and PL/SQL.

Overall we find that an API approach is a flexible way to integrate IT systems. Further, it complements the GUI provided with the dedicated IT system and allows the shipyard's IT department to automate tasks performed frequently.

3.1.2. View Approach

If the data in the data storage is made available to other IT systems via a set of SQL views we call it the view approach. In that case the views can be accessed via the standard SQL query language. The view approach thus provides set oriented access to the data objects whereas the API provides a one-object-at-a-time access to data. The set oriented SQL approach can be used to integrate to business

intelligence systems, *Torp et al. (2002)*. In the view approach, System A accesses the data stored in System B via the views created on System B's data storage, Fig.8. SQL commands are executed in the integration of System A and System B.

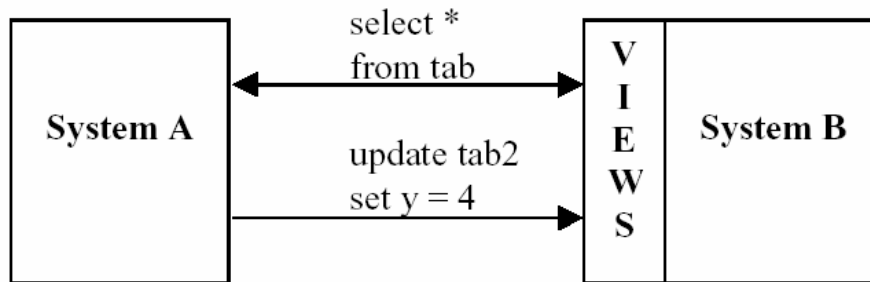


Fig.8: View approach, *Torp et al. (2002)*

With the view approach we only reveal the same information as available via the public attributes of the API approach. It is possible to optimize the data to be in a highly normalized form in the data storage, with all the associated benefits (such as avoiding replication of data and guaranteeing good performance), *Silberschartz et al. (2001)*, while presenting the data meaningfully to the users in a non-normalized form via the views. The views can be made updateable via advanced database features such as instead-of-triggers and the code base provided by the API approach.

The view approach is most well suited for the exchange of small to medium size objects. We believe that the data objects exchanged via the view approach will in general be bigger than the data objects exchanged with the API approach as this approach makes it more natural to update a single attribute, e.g., setting a status flag. In the view approach this is most naturally replaced by fetching an entire row or set of rows. The view approach allows for a very high number of exchanges as it can be build on highly scalable database systems, *Torp et al. (2002)*.

The standard SQL query language applied by the view approach is a very widely used standard and provides a good basis for the integration. Most database systems also allow access to distributed data, which makes the approach well suited for both internal and external integration. We believe that the view approach and the pull principle go well together, allowing other IT systems to query and retrieve the information they need to integrate to another IT system.

3.2 Functional Integration

The functional integration of planning an resource management has his starting point with the definition and scheduling of processes. Hence the activity network builds the backbone for the integration. With respect to the basic principles mentioned above this has to be an hierarchical decomposition of processes starting on rough level (Rough Planning). The vertical integration is via an hierarchical network approach. I.e. a rough planning activity is the input for the detailed planning. It provide the boundaries for the decisions on shop floor level by giving a time frame and budget of hours (or costs).

The resource management is integrated as follows: All resources needed to perform the process are linked to the activities. This can be done on any level in the hierarchy. This reference is available in the different functions.

In the following we will give some examples for the integration of planning and resource management. We will focus on the resources 'material' (project components), 'information' (drawings), and facilities (floor space).

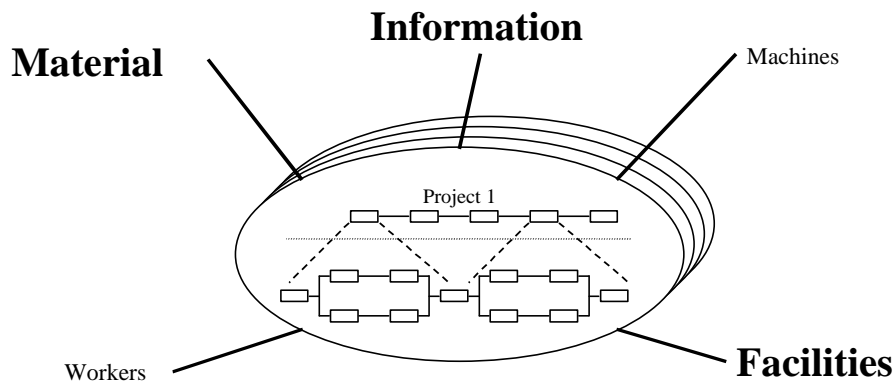


Fig.9: Integrated Resource Management

3.2.1. Project Components

With respect to material purchasing – especially of long-lead items – it is essential to be able to calculate requirement date and related purchase order date in the Master Planning phase already. Those dates have to be updated in case of changes. Long-lead items, available in the Material Management System (as Project Components) can be attached to the planning activities. The linkage to activities provides the long lead items with requirement dates via the scheduling of the Master Plan Project. This is equal to the calculated start date of the activity.

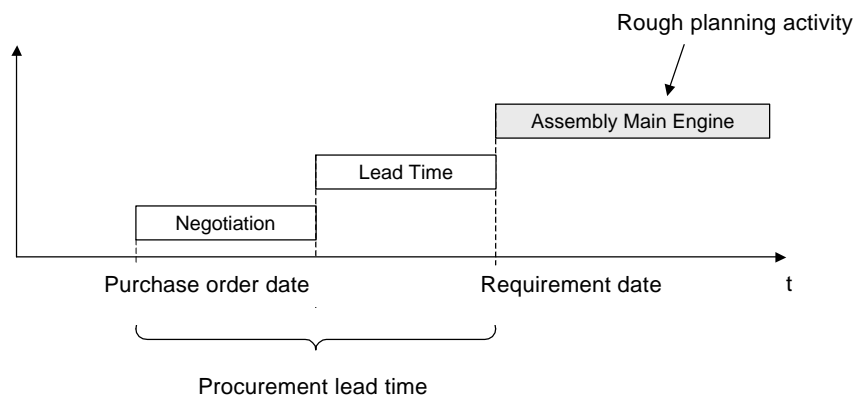


Fig.10: Procurement schedule

In the course of further detailing of the network in subsequent planning steps, the user can re-allocate materials to activities of lower project level. Material receive updated (detailed) requirement dates in that way. In case of rescheduling of the network, the requirement dates of components are updated automatically. The result is the list of long-lead items to be purchased and the respective dates:

- deployment per,
- specification needed per,
- purchase per,
- start negotiation per.

This input is going to purchasing either directly or via the Material Management System. The feedback from Material Management with actual dates is reflected in the planning system to indicate problems.

3.2.2. Drawings

The flowchart in Fig.11 illustrates the planning of drawings in relation to activities and resources.

(1) In Rough Planning an activity is created. This is basically a normal activity marked as ‘Design’ which is linked to a respective production activity (2). This is done to define the due date for Design, i.e. to specify the date at which the drawings are required. To specify which department, group, external company, etc. has to create the drawings a resource requirement is attached to the design activity (3). (4) Drawings, in the meaning of creating an entry in the drawing catalogue with a drawing id, are created in the CAD-system or MARS depending on the origin. The created Design-Activities from Planning are available in MARS and are attached to the relevant drawings (5). From now on the planned issuance dates for the drawings are taken from the schedule from planning. (6) Next step is to split up the activity defined in ROUGH PLANNING into single jobs, usually one per individual drawing. (7) Assigning an individual drawing from the set of drawings defined in ROUGH PLANNING to a Detailed-Activity changes the input for the drawing management as well. The drawing catalogue is available in Planning System. The final task is assigning an individual resource (group or person) to a Detailed-Activity (8). This is done to define which person is responsible for a drawing and when it has to be delivered.

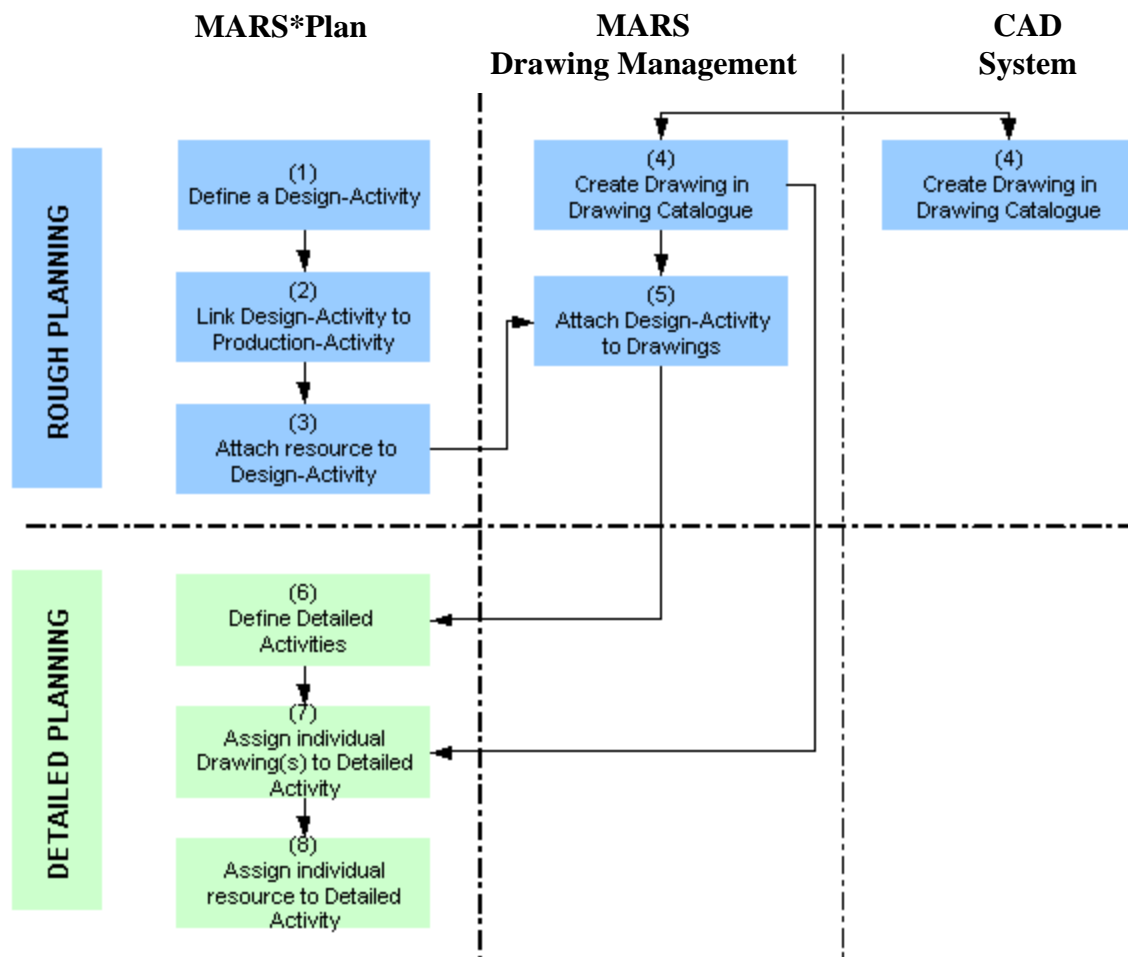


Fig.11: Integrated Drawing Management

3.2.3. Floor Space Management

For actual floor space allocation, one must check whether the Production Program is feasible with respect to floor space needed. Within order processing in shipbuilding floor space for assembly and erection often forms a bottleneck. Therefore it has to be considered as limiting resource during scheduling and resource planning similar to labor and machinery. The floor space management

module (SpaceMan) supports the allocation of floor space with objects. The system performs simultaneously collision check:

- geometry,
- geometry over time,
- weight constraints,
- height constraints

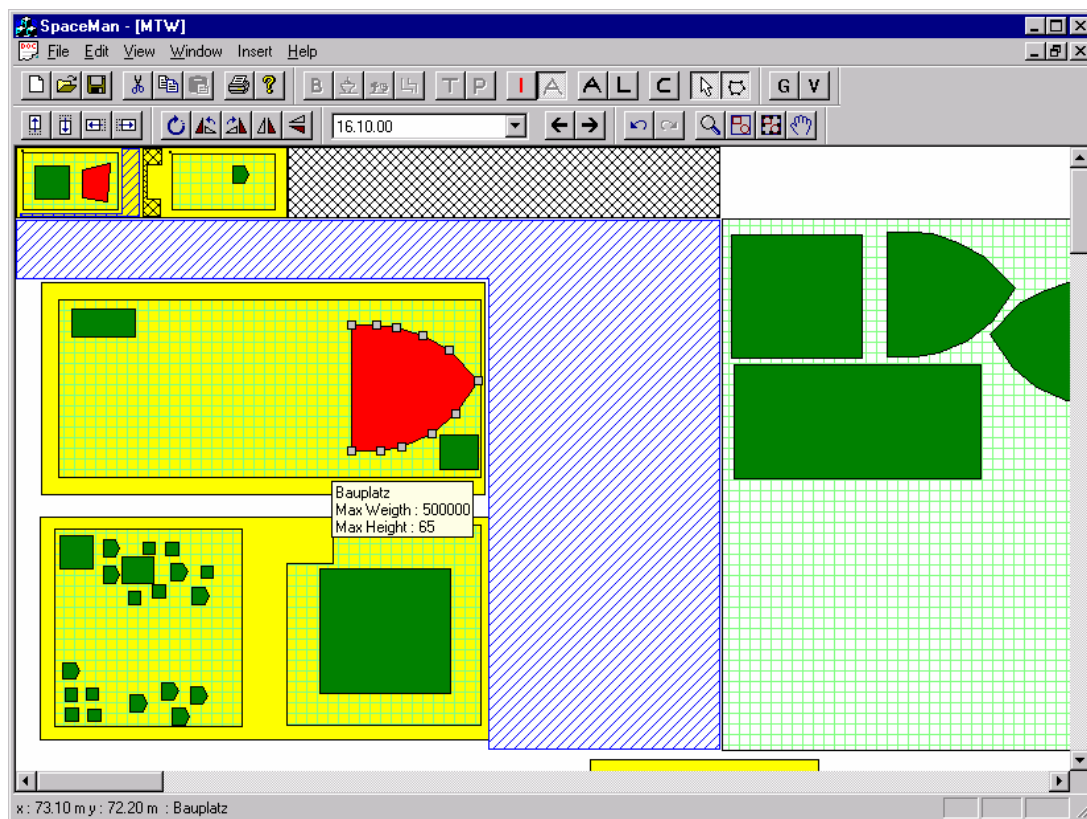


Fig.12: Floor Space Management

A substantial aspect of the floor space allocation scheme is the consideration of the time as dimension of the planning problem. The linkage of objects to activities in the planning system supports to use the activity's dates for the definition of the required floor space. Events that change the dates of a linked activity could triggers collision. This is checked for the activities of all project (versions) belonging to the selected scenario. Result of this process is the Space Plan on the one hand and the feasible Production Program on the other.

4 Conclusion

Shipbuilding as an extremely complex process requires special methods and tools for order processing. Since no monolithic solution is able to answer the requirements "Best of Class" solutions must be integrated. The integrating element for the functional integration of planning and resource management is the activity network. For which a hierarchical approach is required to integrate planning horizontally and vertically.

For the technical integration of dedicated software systems we recommend either API or View Approach depending. Some examples of integration of computer supported planning and resource management have been explained. They show that one can improve the interaction of different business processes with relatively low effort just by establishing logical links in the system architecture.

References

SIKSNE-PEDERSEN, I. (2000), *Material management in shipbuilding*, European Shipbuilding in the 21st Century, London

TORP, K.; RIISBERG, L.; BØRGLUM, L.; (2002), *Integrating logistic systems in shipbuilding*, 11th Int. Conf. Computer Applications in Shipbuilding, Malmö

SILBERSCHARTZ, A.; KORTH, H. F.; SUDARSHAN, S. (2001), *Database system concepts*, McGraw-Hill, ISBN 0-07-228363-7

Artificial Neural Networks for Hull Resistance Prediction

Patrick Couser, Andrew Mason, Formation Design Systems. (patc, andym,
@formsys.com)

Garth Mason, Cameron R. Smith, Brian R. von Kinsky, Curtin University of
Technology. (masongc, smithcr, bvk @cs.curtin.edu.au)

Abstract

The applicability of artificial neural networks to the problem of ship resistance prediction as an alternative to more traditional statistical regression models has been investigated. In this work, an artificial neural network has been used as an interpolation tool to predict the residuary resistance of a systematic series of catamaran forms. It has been found that an artificial neural network is able to produce results of sufficient accuracy to be useful for preliminary prediction of vessel resistance, with the major benefits of: being relatively simple to set up; being easily retrained with new data; and that Froude number may be easily included as an independent variable.

Nomenclature

A	Static wetted surface area
B	Demihull maximum beam on waterline
C_F	Coefficient of frictional resistance [ITTC-57 Correlation line]
C_R	Coefficient of residuary resistance
C_T	Coefficient of total resistance
C_W	Coefficient of wave resistance
C_{WP}	Coefficient of wave pattern resistance
F_n	Froude Number, $[v/\sqrt{gL}]$
g	Acceleration due to gravity
L	Vessel length between perpendiculars
$L/\nabla^{\frac{1}{3}}$	Slenderness ratio
Re	Reynolds Number, $[vL/\nu]$
S	Catamaran demihull centreline separation
T	Demihull draught
v	Vessel velocity
∇	Demihull volume of displacement
ν	Fluid kinematic viscosity
ρ	Fluid density

(ITTC Resistance Coefficients = resistance/ $\frac{1}{2}\rho Av^2$)

1 Introduction

The aim of this research was to determine whether artificial neural networks can be used as an alternative to traditional statistical regression techniques for the interpolation of vessel resistance from tank test results.

For some time, artificial neural networks have been used in many diverse fields as function approximation tools, producing results comparable with (or better than) regression models and other statistical methods, see for example *Negnevitsky (2001)* or *Neocleous and Schizas (1995)*. One of their key advantages is their ability to easily model complex, non-linear systems, a feature which is not true of statistical regression methods where an appropriate non-linear function must first be found.

Table 1: Hull model notation and main parameters

$L/\nabla^{\frac{1}{3}}$	B/T		
	1.5	2.0	2.5
6.3	—	3b	—
7.4	4a	4b	4c
8.5	5a	5b	5c
9.5	6a	6b	6c

In the field of Naval Architecture, interpolation and prediction of hull resistance from model experiments and tank testing has traditionally used statistical regression methods. However, this is a problem that is also well suited to the application of artificial neural networks. Specifically, artificial neural networks may offer a more favourable solution than statistical methods due to greater flexibility and the ease with which they can be applied to complex non-linear systems, see for example *Jain et al. (1996)*.

An advantage of artificial neural networks over statistical methods is their ability to adapt to new data. Once an artificial neural network architecture has been designed, it can quickly be retrained as new data becomes available. In addition, the same artificial neural network topology can be trained with other data for different vessel types. Retraining the artificial neural network with additional data is generally simpler and quicker than recomputing a statistical model.

Finally, the difficulty of using statistical approaches for prediction escalates quickly as the number of inputs (independent variables) or non-linear nature of the data increases, whereas the difficulty of developing a neural network increases less dramatically with the complexity of the system.

A disadvantage of artificial neural networks is the lack of a single approximation equation: the result is an artificial neural network topology and set of weights. This means that the prediction can only be calculated on a computer with suitable artificial neural network software. In practice this is not really a big disadvantage because regression equations can also only really be developed and implemented with suitable software.

It is also true that it is usually a goal for statistical regression models to be based on an equation which has some physical significance. In some cases the physical significance of the regression equation is subject to interpretation. However, an artificial neural network does not represent a physically meaningful model of the data, which could be considered a disadvantage.

1.1 Resistance data

The resistance data used for this investigation was from a series of resistance experiments carried out on a systematic series of hull forms tested in both monohull and catamaran configurations. These experiments together with the resistance data are described in full in *Molland et al. (1994, 1995)* and *Insel and Molland (1992)*. A typical body plan of these hulls is shown in Figure 1 and the range of hull parameters tested is summarised in Table 1. A total of ten hull forms were tested, covering a slenderness ratio range of $L/\nabla^{\frac{1}{3}} = 6.3$ to 9.5 and a Breadth:Draught ratio range of $B/T = 1.5$ to 2.5. In addition, all models were tested as monohulls and in catamaran configurations at Separation:Length ratios of $S/L = 0.2, 0.3, 0.4$ and 0.5. Most configurations were tested in the Froude number range $F_n = 0.2$ to 1.0. However, for models 3b and 4b (see Table 1), at the closer demihull catamaran configurations, the upper F_n tested was reduced due to excessive spray between the demihulls.

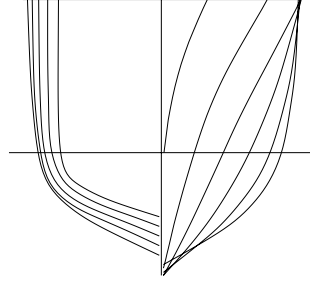


Figure 1: Typical hullform tested ($B/T = 2.0$)

In the present work, only modelling and function approximation of residuary resistance (C_R) was considered – see Equation 1. Additional data such as wave pattern resistance coefficient (C_{WP}), running trim and sinkage were also measured during the experiments and these could be modelled using the same artificial neural network topology retrained with the relevant data.

$$C_R = C_T - C_F \quad (1)$$

where C_F is calculated from the ITTC-57 correlation line: $C_F = \frac{0.075}{(\log R_e - 2)^2}$

2 Development of artificial neural network architecture and training

2.1 Selection of input parameters and output values

One of the key factors in the development of a successful predictive tool (irrespective of whether the method is statistical regression or an artificial neural network) is the selection of appropriate input parameters (or independent variables); it is important to include only those parameters that have a significant influence on the value of the predicted result.

The principal parameters that were varied in the catamaran systematic series were: $L/\nabla^{\frac{1}{3}}$, L/B , B/T and S/L , and the hulls were tank tested over a range of F_n . In the original analysis of the tank results (*Molland et al. 1994*), a very high degree of correlation between the effects of $L/\nabla^{\frac{1}{3}}$ and L/B was observed. For this reason, it was decided that the only input parameters that should be used were: $L/\nabla^{\frac{1}{3}}$, B/T , S/L and F_n .

One problem often found with statistical regression methods is the highly non-linear relationship between resistance and Froude number (illustrated by the typical resistance curve in Figure 2). This can cause difficulty in obtaining a regression equation with Froude number as an independent variable or, as is more often the case, quite a large number of independent regression equations are developed for individual Froude numbers. A benefit which will be seen with artificial neural networks is their ability to model the non-linear relationship between resistance and Froude number and thus only require one artificial neural network instead of different ones for each individual Froude number.

Some regression methods have attempted to include F_n explicitly as an independent variable. However, it is often still not possible to have a single equation because of the complex nature of the variation of C_R with F_n . For example, *Holtrop (1977)* includes F_n as an independent variable but requires two equations: one for slow speeds, below the resistance hump, and a second for higher speeds.

2.2 Development of the artificial neural network architecture

Selection of a suitable artificial neural network architecture is probably the hardest part of the problem and critical to obtaining a useful artificial neural network. It is analogous to selecting the

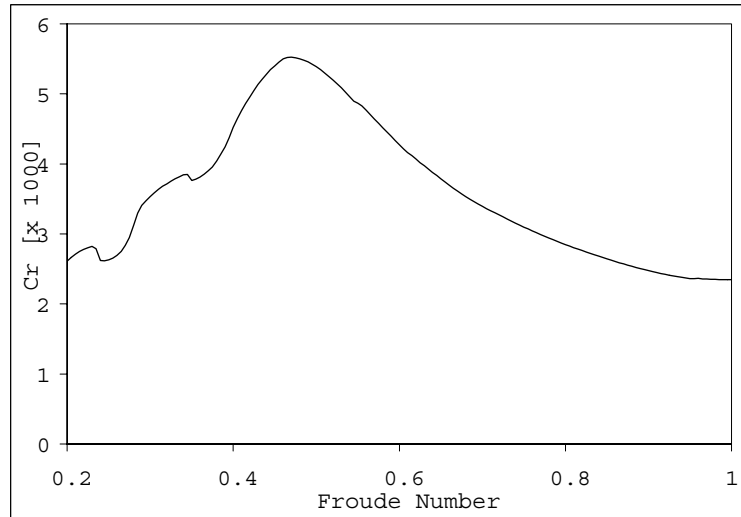


Figure 2: Typical residuary resistance coefficient curve

form and independent variables of a regression equation. One of the advantages of the artificial neural network approach is that it is easy to automate searching for an optimum architecture, either by using a simple exhaustive search or by using some sort of heuristic search methodology.

Alyuda's NeuroIntelligence software (*Alyuda 2004*) was used for the artificial neural network calculations. This proved to be a very powerful tool for carrying out the artificial neural network design and analysis.

During the development of the artificial neural network design, the available input data was split into three, mutually exclusive, sets described below. In all cases, a 70%, 15%, 15% split between the training, validation and test sets was used.

Training set: used to train the artificial neural network, i.e. to adjust the artificial neural network weights to maximise the artificial neural network's predictive ability and minimise its forecasting error.

Validation set: used to tune the artificial neural network topology or parameters other than weights. Also used for automatic comparison of alternative artificial neural networks.

Test set: used *only* to test the accuracy of the predictions made by the artificial neural network on new data.

The accuracy of the artificial neural networks can be measured in several ways:

Test error: the mean absolute artificial neural network error on the test set. The lower the value, the better the artificial neural network is at predicting the test set.

Fitness: in the work presented here, the fitness is based on the inverse of the test error, thus the greater the fitness, the better the artificial neural network.

R-squared: a standard statistical measure. R-Squared = 1 implies a perfect fit.

Correlation coefficient, r: the correlation between the test set and the predicted values from the artificial neural network. $r = 1$ implies a perfect fit.

Akaike's criterion: a measure of the error on the test set compared with the complexity of the system; systems with greater complexity are penalised. The greater the number, the better the artificial neural network.

Of the measures available for comparing the accuracy of the alternative artificial neural networks, the fitness score based on test error was found to be the most sensitive and was used to select the best artificial neural network architecture. Akaike's criterion was also found to be a useful quantity for comparing artificial neural networks with a single hidden layer and both R-Squared and the correlation coefficient tended to be maximised for the artificial neural networks with the highest fitness scores based on test error and Akaike's criterion.

The accuracy of the predicted results from the artificial neural networks was also assessed in a qualitative manner by visual inspection of the resistance curves predicted by the artificial neural networks and their ability to interpolate results for input values between those actually tank tested.

The training algorithm used to train the artificial neural network was found to have quite an important influence on its accuracy and the speed with which the training converged (or whether it converged at all). In general, the artificial neural network starts with random weights and the training process adjusts these weights with the aim of producing an accurate prediction of the training data. Because this is a semi-random process (due to the initial values of the weights), it is important to retrain the artificial neural network several times with different starting values for the weights. It is also important to allow sufficient iterations of the training regime to allow the artificial neural network to converge. The required computational effort increases considerably as the number of neurons in the artificial neural network, the number of retrainings and the number of allowed training iterations increase. Because of limited computational resources a compromise is required between these factors. It is for these reasons that it is not possible to perform an exhaustive search of all possible artificial neural network architectures and that there can be considerable scatter in the fitness results for alternative artificial neural network architectures.

2.3 Fitting single resistance curve

In order to determine how well an artificial neural network could fit a typical C_R vs. F_n curve (Fig.2), a simplified problem with only one hull model was investigated. This data set had only one input variable (F_n). Artificial neural networks with up to two hidden layers were investigated, the number of neurons in these layers was systematically varied.

Initially, a wide range of neurons in a single hidden layer artificial neural network was investigated. It was found that a minimum of six neurons in the hidden layer were required to provide a reasonable approximation to the shape of the C_R curve, whilst around 10 to 20 neurons provided the most accurate fit.

Fig.3 shows how the various fitness criteria varied for the single hidden layer artificial neural network as the number of hidden layer neurons was increased. In this figure, the fitness is based on the inverse of the test error. Hence, the greater the fitness, the better the artificial neural network was able to predict the training set. The number of hidden layer neurons was varied between one and 50.

Fig.4 shows the effect of varying the number of hidden layer neurons on the predicted resistance curve compared with the actual C_R curve. It can be seen that the accuracy of the fit increases quickly with increasing number of hidden layer neurons. This is because the artificial neural network has a greater number of degrees of freedom and is better able to model the non-linear nature of the resistance curve. As the number of neurons was increased further, the accuracy of the network was found to deteriorate. This was because although the training set was modelled

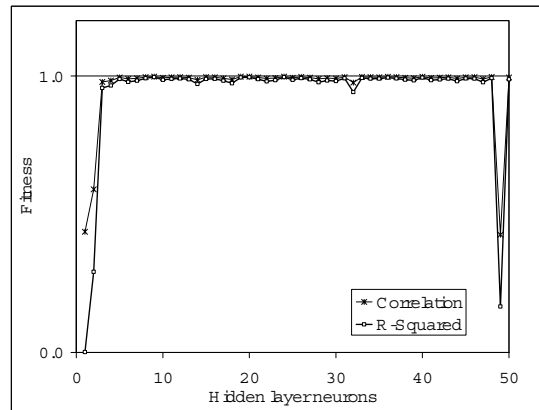
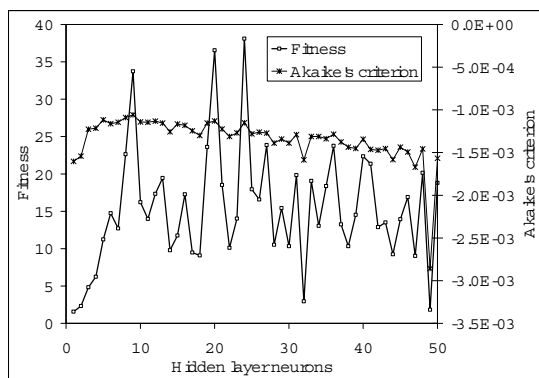


Figure 3: Influence of number of hidden layer neurons on artificial neural network accuracy (fitness score) for an artificial neural network with a single hidden layer

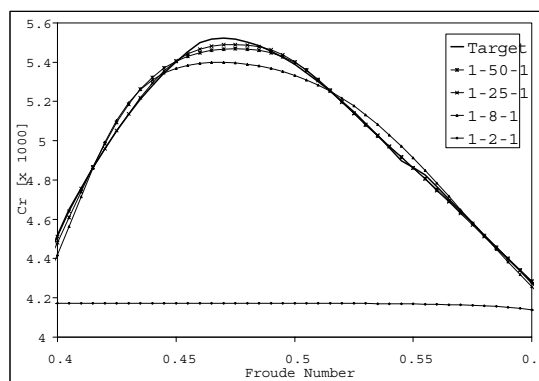
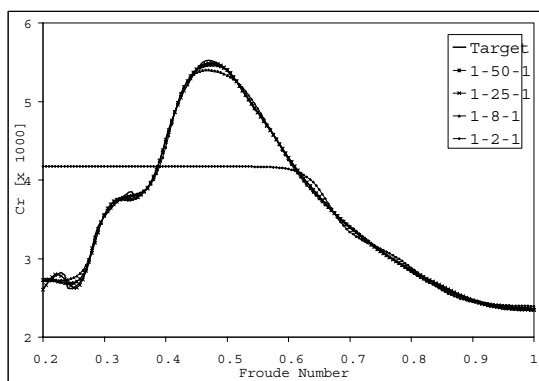


Figure 4: Influence of number of hidden layer neurons on artificial neural network accuracy (prediction of resistance curve) for an artificial neural network with a single hidden layer

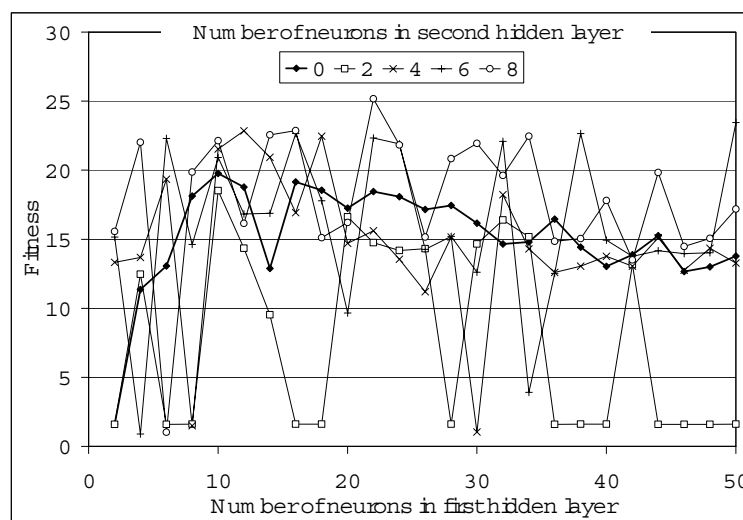


Figure 5: Influence of number of hidden layer neurons on artificial neural network accuracy (fitness score) for artificial neural networks with single and two hidden layers

more accurately, a degree of over fitting had occurred and the test set was not well predicted (see Section 2.5 for further discussion of over fitting.)

2.4 Number of hidden layers and hidden layer neurons

Some investigations into artificial neural networks with two hidden layers were performed, but the results were somewhat inconclusive. Fig.5 shows the effect of increasing the number of hidden layers to two. Fitness curves are plotted against the number of neurons in the first hidden layer for different numbers of neurons in the second layer. The number of neurons in the first layer was varied between two and 50, whilst the number in the second layer was varied between two and eight. (The artificial neural network with a single hidden layer is shown in the bold line, with the solid diamonds, whilst the artificial neural networks with two hidden layers are shown by the lighter lines with open symbols or crosses.) It can be seen that adding a second hidden layer did not significantly improve the artificial neural network's performance and increasing the number of neurons in the second hidden layer had little effect. In fact, for the artificial neural network with two neurons in the second hidden layer, there are many cases where the training failed to converge. The only cases where two hidden layers were possibly more effective was where there were few (less than ten) neurons in the first hidden layer. The large degree of scatter in the fitness values has been discussed at the end of Section 2.2.

The observations of *Neocleous and Schizas (1995)*, which indicate that rarely are multiple hidden layers effective in terms of both accuracy and speed of training, are borne out.

It is for the reasons described above that an artificial neural network architecture with a single hidden layer was selected.

A search of the available literature suggested many (sometimes conflicting) rules of thumb for determining the number of neurons in the hidden layer. It became apparent that there is no general rule by which a suitable number of hidden layer neurons can be chosen and that the optimum number is very specific to the problem being solved.

For example, Kolmogorov's theorem *Brattka (2003)* suggests that the optimum number of hidden layer neurons is $2n+1$, where n is the number of inputs to the artificial neural network. However, the number of hidden layer neurons suggested by the NeuroIntelligence software is half this number. For this particular problem, it is thought that the high degree of non-linearity in the C_R vs. F_n curve requires quite a large number of neurons. The non-linear relationship between C_R and F_n is in contrast to the relationship between the other independent variables ($L/\nabla^{\frac{1}{3}}$, B/T and S/L) which were only varied over a range of three to five values, and were often nearly linear functions. The relatively large number of hidden layer neurons required to fit the C_R vs. F_n curve may cause over-fitting to the remaining independent variables. This is discussed further in the following section.

2.5 Over-fitting

Both statistical methods and artificial neural networks can suffer from over-fitting. Over-fitting occurs as the number of degrees of freedom, or unknowns, approaches the number of training data points. If over-fitting occurs, the training data is predicted very well, but new input data is often poorly predicted.

The method used to overcome this is to divide the data into mutually exclusive training and testing sets. The training set is used to train the artificial neural network (or solve the regression equation) and the test set is used to test the accuracy of the fit. Since the test set is not used in the training, any over-fitting to the training set will lead to a poorer result when using the test set and a lower fitness score for that particular artificial neural network. As mentioned in Section 2.2, an additional validation set is also sometimes used.

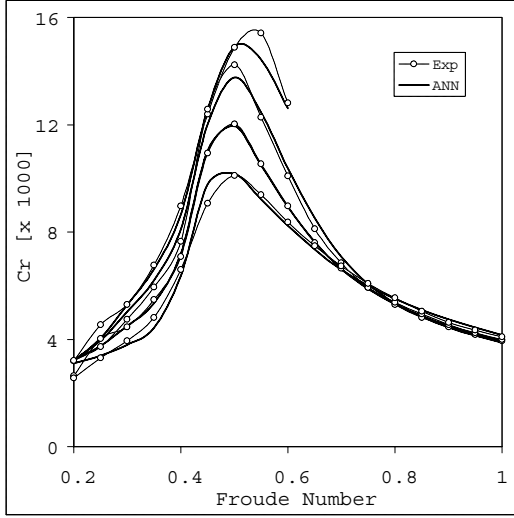


Figure 6: Model 3b

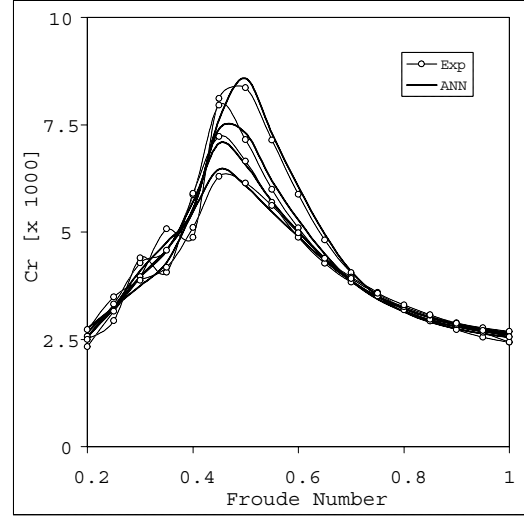


Figure 7: Model 4a

In the problem presented in this paper, the data is highly non-linear with respect to Froude number but is quadratic with B/T (three points tested) and cubic with $L/\nabla^{\frac{1}{3}}$ and S/L (four points tested). Hence, for a statistical regression model, one might choose a function which includes four or five (or more) non-linear functions of F_n but probably only a linear function of B/T and perhaps up to a quadratic function of $L/\nabla^{\frac{1}{3}}$ and S/L ; a total of around 13 unknowns. With an artificial neural network, it is not possible to be so specific about how the function varies with each independent variable. It is simply a matter of choosing a suitable number of neurons. However, the number of neurons must be sufficient to have a good fit with respect to F_n but this can lead to an over-fitting with respect to B/T , $L/\nabla^{\frac{1}{3}}$ and S/L . Hence with the artificial neural network problem, over-fitting can only be avoided by having a sufficiently large test set. The trade-off is that increasing the size of the test set reduces the quantity of data in the training set and hence (in general) the accuracy of the fit.

2.6 Artificial neural networks for catamaran systematic series resistance data

Further investigations into suitable artificial neural network architectures for modelling the resistance data of the whole systematic series indicated that it was not necessary to increase the number of hidden layers or neurons compared with the artificial neural network used to predict the resistance curve for a single vessel. This is thought to be because the artificial neural network has sufficient degrees of freedom to model the relatively linear variation with the other independent variables: $L/\nabla^{\frac{1}{3}}$, S/L and B/T . The final artificial neural network architecture selected had a single hidden layer with 15 neurons.

3 Results

Results from the artificial neural network described in the previous section are presented in Figures 6 to 15. These figures compare the resistance data from experiments (used as the training data for the artificial neural network) with the predictions from the artificial neural network. (The results from experiments are shown as open circles and the predicted data from the artificial neural network as a solid line.) In each figure, the results for the four catamaran configurations tested are presented, the closest spacing ($S/L = 0.2$) causing the greatest resistance hump and the widest spacing ($S/L = 0.5$) causing the smallest. It can be seen that there is generally very good agreement for all the hulls and conditions tested.

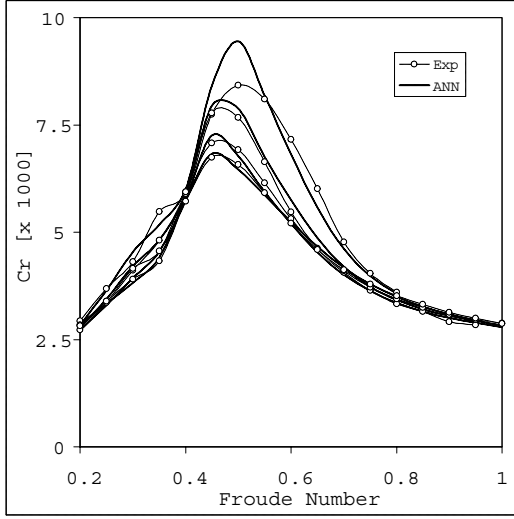


Figure 8: Model 4b

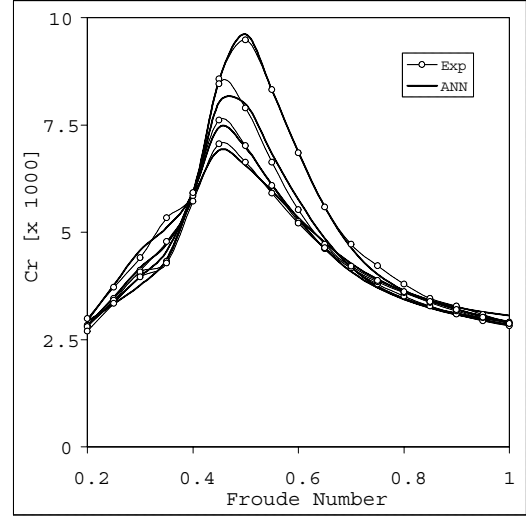


Figure 9: Model 4c

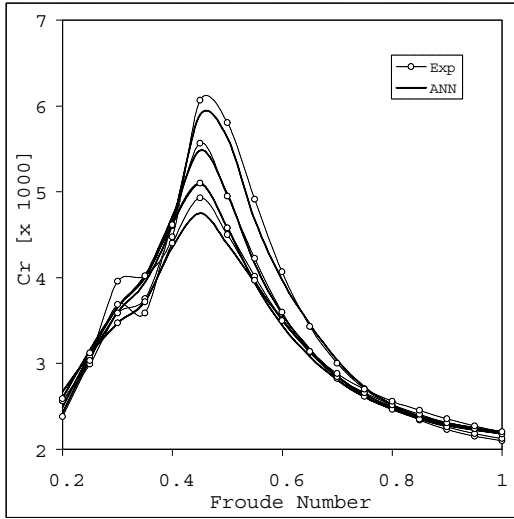


Figure 10: Model 5a

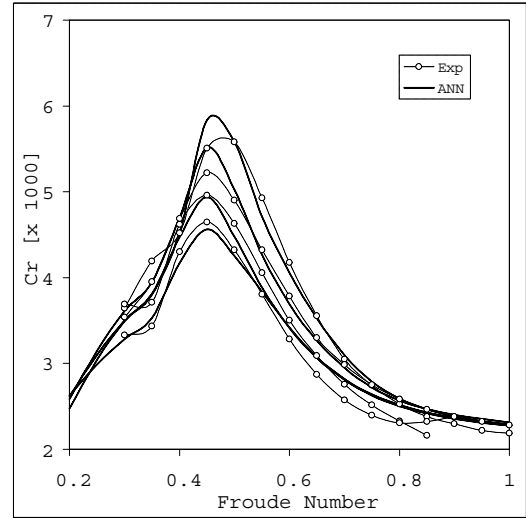


Figure 11: Model 5b

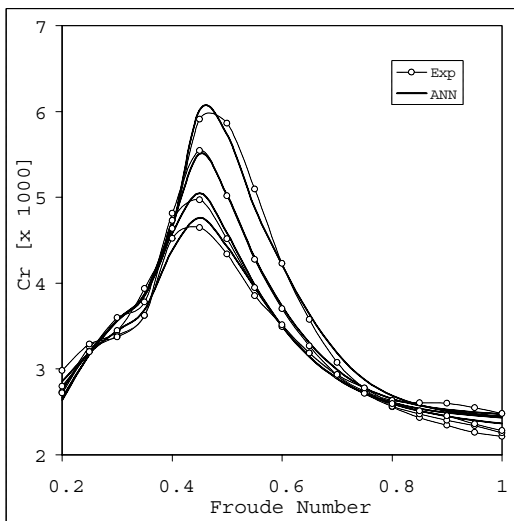


Figure 12: Model 5c

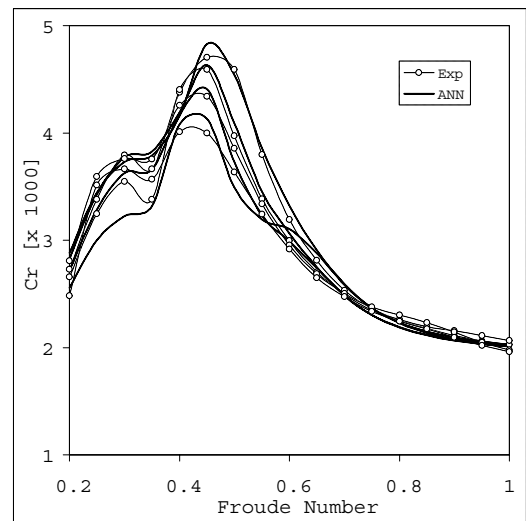


Figure 13: Model 6a

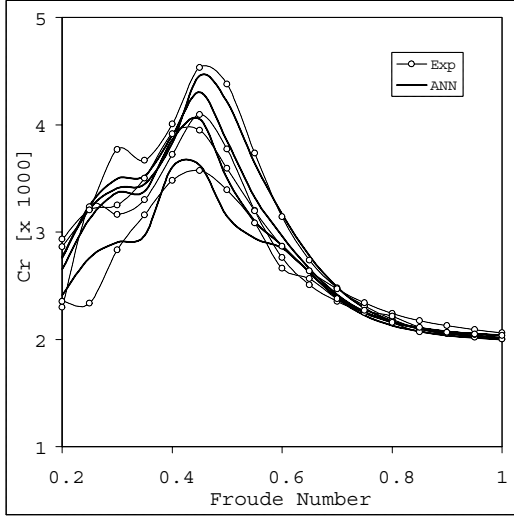


Figure 14: Model 6b

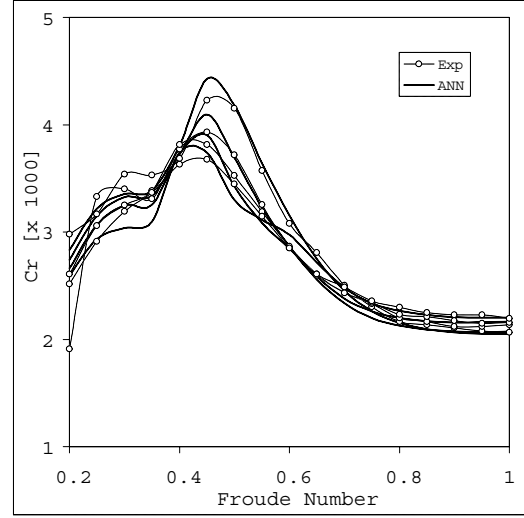


Figure 15: Model 6c

3.1 Interpolation and extrapolation

In order to assess the real usefulness of the artificial neural network, it has been used to interpolate data between the tested values of S/L , B/T and $L/\nabla^{\frac{1}{3}}$. The input parameter values have also been extended beyond the range of the original training set to examine the ability of the artificial neural network to extrapolate data – something which can be quite dangerous with statistical regression formulae.

Figure 16 shows the C_R variation with S/L for a catamaran with demihulls with $L/\nabla^{\frac{1}{3}} = 8.0$ and $B/T = 2.0$. Figure 17 shows the effect on C_R of varying demihull $L/\nabla^{\frac{1}{3}}$ for a catamaran with $S/L = 0.3$ and demihull $B/T = 2.0$. Finally, Figure 18 shows the effect of B/T variation for a catamaran with $S/L = 0.3$ and demihull $L/\nabla^{\frac{1}{3}} = 8.0$. It can be seen that, in all cases, the artificial neural network produces smooth response surfaces which behave well even when extrapolating beyond the range of the training data.

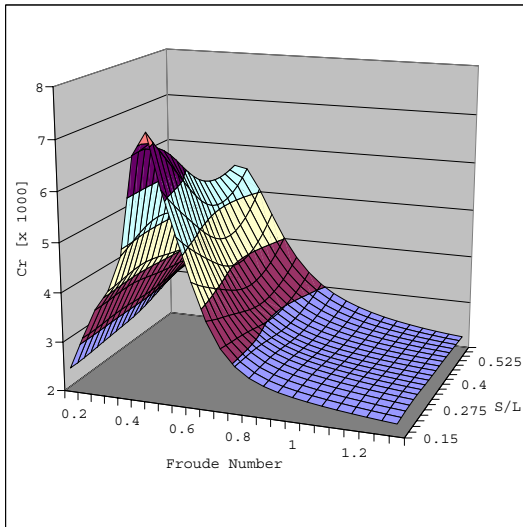


Fig.16: Effect of S/L for a demihull with $L/\nabla^{\frac{1}{3}} = 8.0$ and $B/T = 2.0$

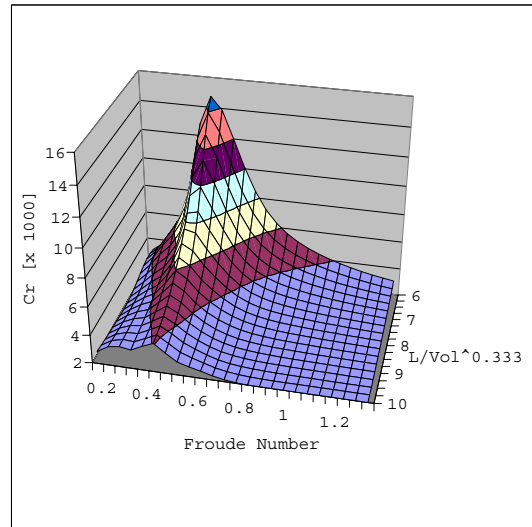


Fig.17: Effect of $L/\nabla^{\frac{1}{3}}$ for a cat. with $S/L = 0.3$ and demihull $B/T = 2.0$

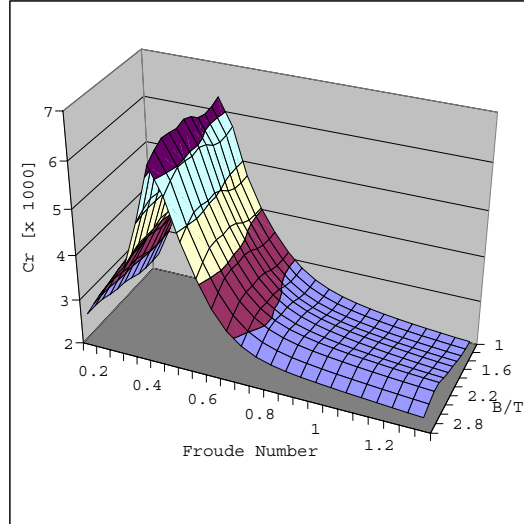


Figure 18: Effect of B/T for a catamaran with $S/L = 0.3$ and demihull $L/\nabla^{\frac{1}{3}} = 8.0$

4 Conclusions

It has been shown that artificial neural networks are able to provide good approximations to hull resistance data and that a single artificial neural network may be used with Froude number as an independent variable over a wide speed range.

An artificial neural network with a single hidden layer and 15 neurons in the hidden layer was found to adequately model the residuary resistance of a systematic series of catamarans where slenderness, Breadth:Draught and Separation:Length ratios were varied for a geosim hull form. Further, the artificial neural network was sufficiently well behaved so as to allow extrapolation outside the range of test data for all independent variables. The single hidden layer architecture is recommended, since addition of further hidden layers did not appear to improve the accuracy of the artificial neural network and only added to the artificial neural network's complexity and required training time; in some cases preventing the training algorithm from converging.

The use of automated artificial neural network software makes it very quick and easy to produce and train an artificial neural network capable of modelling hull resistance problems. This is considerably easier and quicker than traditional statistical methods. Whilst it is important to choose a reasonable artificial neural network architecture, the exact number of neurons in the hidden layer is not too critical.

5 Further work

The artificial neural network approach makes it very easy to either generate a new artificial neural network for another systematic series or to add more data to the existing one. It is simply a matter of retraining it with the additional data. This will be done to produce artificial neural networks to predict C_{WP} , trim and sinkage data also measured by *Molland et al. (1994)*.

References

- ALYUDA (2004), *NeuroIntelligence manual*, <http://www.alyuda.com>.
- BRATTKA, V. (2003), *A Computable Kolmogorov Superposition Theorem*, <http://www.informatik.fernuni-hagen.de/thi1/vasco.brattka/publications/kolmogorov.html>.
- HOLTROP, J. (1977), *A Statistical Analysis of Performance Test Results*, International Ship-building Progress.
- INSEL, M.; MOLLAND, A. (1992), *An Investigation into the Resistance Components of High Speed Displacement Catamarans*, Transactions of the Royal Institution of Naval Architects, 134(A).
- JAIN, A.; MAO, J.; MOHIUDDIN, K. (1996), *Artificial neural networks: A tutorial*, IEEE Computer, 29(3):31–44.
- MOLLAND, A.; WELLICOME, J.; COUSER, P. (1994), *Resistance Experiments on a Systematic Series of High Speed Displacement Catamaran Forms: Variation of Length-Displacement Ratio and Breadth-Draught Ratio*, Ship Science Report 71, Department of Ship Science, University of Southampton.
- MOLLAND, A.; WELLICOME, J.; COUSER, P. (1995), *Resistance experiments on a systematic series of high speed displacement catamaran forms: Variation of length-displacement ratio and breadth-draught Ratio*, Transactions of the Royal Institution of Naval Architects, 138(A)
- NEGNEVITSKY, M. (2001), *Artificial Intelligence: A Guide to Intelligent Systems*, Addison Wesley
- NEOCLEOUS, C.; SCHIZAS, C. (1995), *Artificial neural networks in marine propeller design*, ICNN'95 – Int. Conf. on Neural Networks, Vol.2, pp.1098-1102, New York, NY, IEEE Computer Society Press

Automation Tools in the Design Process

Reidar Tronstad, SENER Ingeniería y Sistemas, S.A., reidar.tronstad@sener.es
Antonio Rodriguez, SENER Ingeniería y Sistemas, S.A., antonio.rodriguezg@sener.es

Abstract

The design process is supported by the use of several technologies, the most important of which is the correct use of a CAD System. But even shipbuilding CAD Systems are developed according to global requirements which may or may not comply with the specific procedures or “modus operandi” in a particular design office. An emerging trend to face this question is to include in the CAD System certain tools to “program” specific procedures using the internal API’s of the CAD System. This is a delicate question because end users should not be involved in complex development activities and it should not be necessary for shipyards or design offices to have to invest in maintaining a development team. The solution must provide an easy to use, but powerful, environment in which a designer may feel comfortable developing these procedures. This paper sets out to describe the requirements of such an environment.

1. Introduction

The design process in the shipbuilding industry is quite complex and cannot be made sufficiently abstract to comply with all the requirements coming from the heterogeneous community of ship designers. This is specially true in the detail engineering process, where many more specific procedures are applied and different production cultures must be taken into account. Software vendors of CAD Systems for shipbuilding have two different strategies to deal with this problem.

The first one is to develop these requirements upon customer request. This alternative is quite expensive, as software development is a complex business. If the software vendor tries to recover the investment incorporating it in the standard product, two new difficulties may arise. First, other customers may not see the advantage of using such a new functionality. Second and worse, the procedure may be considered as confidential by the original customer making the request.

The second alternative is to provide, with the CAD System, a set of tools for custom development with which the user is able to create these specific procedures. This is the more flexible approach, but it requires much more effort from the customer point of view. The environment in which the procedures are developed, the programming language, the architecture of the applications, etc are critical factors which determine the feasibility of the solution.

The following paragraphs will try to establish the most important requirement for this type of solutions and the benefits which can be obtained from those kind of tools.

2. Analysis of existing approaches

2.1. AutoCAD

AutoCAD is the most well known general purpose CAD application, from Autodesk. For this reason, the approach of AutoCAD to face this question can be considered as a reference model to compare with other approaches.

In order to use the AutoCAD approach as a reference model, the main differences between the functionality targeted by AutoCAD and the functionality required by a shipbuilding designer to automate the design process must be considered.

As AutoCAD is a general purpose CAD application, it is considered as an horizontal platform on which vertical (sector) applications are developed. In this way, AutoCAD is by itself a development platform for specialized CAD applications (architecture, mechanical, electrical,..., even shipbuilding).

Then, the target audience for the AutoCAD customization tools is the software industry devoted to develop specific AutoCAD sector applications, which are distributed and sold as add-on programs to the AutoCAD general platform.

AutoCAD offers two different environments for customization, AutoLISP and Visual Basic for Applications (VBA). Either of these environments can be combined with menus customization in order to add new commands to the standard configuration of AutoCAD.

2.1.1. AutoLISP and Visual LISP

AutoLISP has been around a long time, and it still shows great vigor. AutoLISP and its big brother Visual LISP are supported by all AutoCAD 2000-based products except the LT editions, AutoDesk Inventor, and Revit. AutoLISP has some powerful benefits: it requires no compilers; its syntax is relatively simple; and it can be easily embedded in menu files.

AutoLISP programs are implemented either by running them interactively from an external file or by defining functions that can be called upon later. It has a particular loading order at AutoCAD startup, involving a few key-files.

Originally an add-on program, Visual LISP was integrated into the AutoCAD product line with AutoCAD 2000. Visual LISP enhances AutoLISP's command vocabulary with a host of functions.

More importantly, Visual LISP users benefit from the ability to work with programming code, the security of the code they produce, and the responsive nature of Visual LISP. Visual LISP utilizes an integrated development environment (IDE), shown in Figure 1, to provide a host of functions. The most relevant features of AutoLISP are:

- **Parenthesis Matching.** This is a simple yet necessary function when working with AutoLISP language, especially for new users. It matches up parentheses and can save hours of program debugging.
- **Color Coding of Syntax.** This allows correctly spelled LISP functions to display in blue and incorrectly spelled functions in black, for instance. The color-coded environment makes it much easier to detect errors than a Notepad session.
- **Database Browsing.** This allows the user to directly browse the current drawing's database for header-type objects such as text styles, dim styles, and so on, making it much easier than the TABLESEARCH method.
- **Reactors.** Visual LISP Reactors can trigger programs based on what the user is doing at the moment (for instance, triggering a certain procedure when a user saves a drawing to set layers and variables correctly). The only way to accomplish this in AutoLISP is to redefine the SAVE, SAVEAS, and QUIT functions (relatively messy), but the VLR reactor commands in Visual LISP allow the program to seamlessly detect what the user is doing and react to it.
- **Compilers.** Visual LISP supports compiling LSP code to secure formats, normally the FAS format. By compiling your code, users will be discouraged from tampering with it.

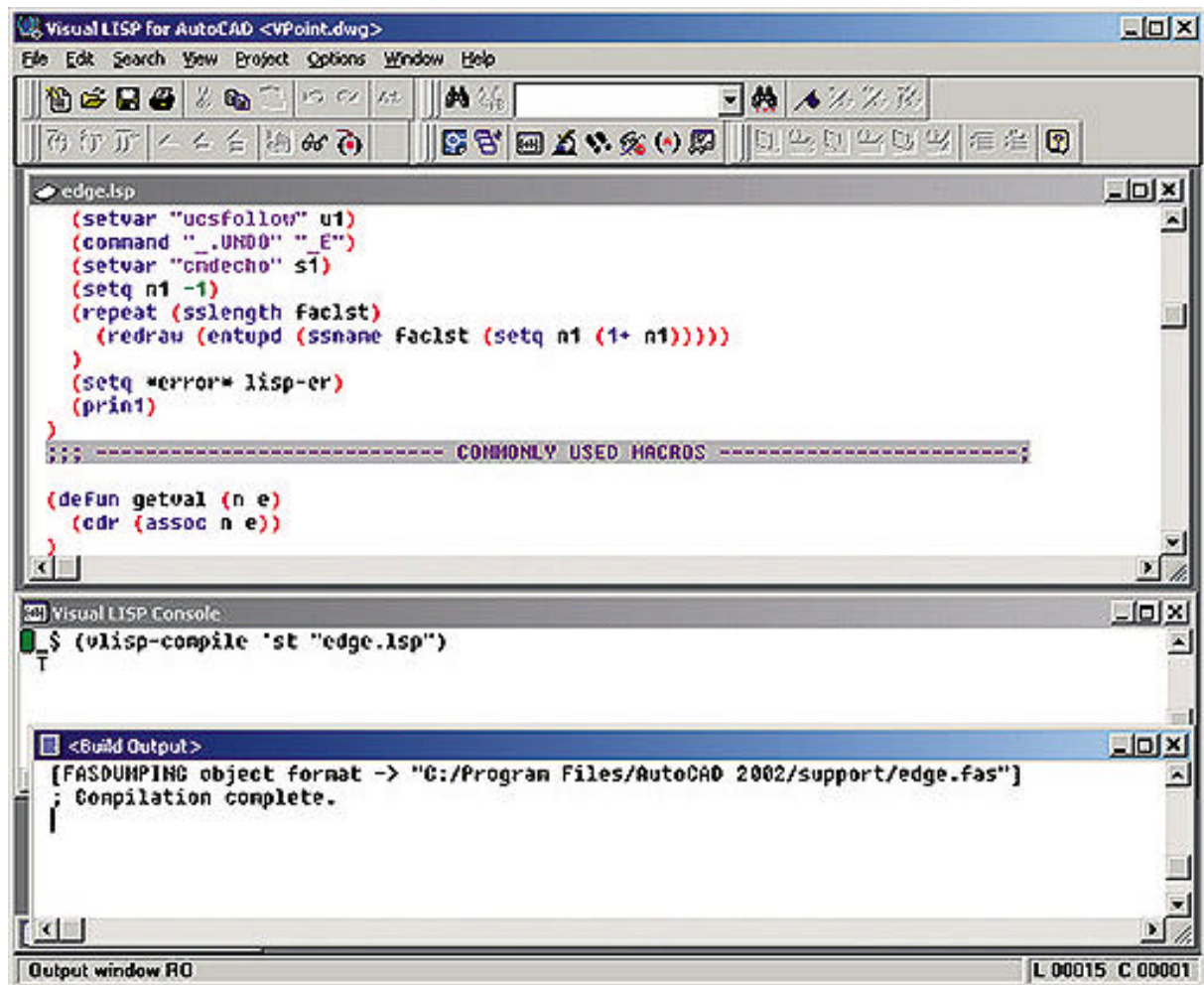


Fig.1: Visual LISP integrated development environment

Visual LISP supports all standard AutoLISP functions, so Visual LISP functions can also be placed in standard LSP files. AutoCAD doesn't really distinguish between the two LISP formats when executing programs. The only limitation of Visual LISP is that it is not a part of R14's default settings, but anyway, Visual LISP is a powerful extension to the AutoLISP platform.

2.1.2. AutoCAD VBA programming

Visual Basic for Applications (VBA) is the newest customization method available in AutoCAD. VBA is a Microsoft technology that was first supported by AutoCAD R14.01. VBA applications do not require a compiler to run inside of AutoCAD. VBA strengths include:

- Creating slick visual forms without having to trudge through DCL-coded dialog boxes, as is the case with AutoLISP.
- Working with data formats other than AutoCAD list types is faster and easier than in AutoLISP.
- Accessing external databases directly, thus bypassing AutoCAD's dbConnect feature set.
- Working with more programmers (anyone wishing to program solutions for AutoCAD must learn the AutoCAD object model--not just any VB programmer can be an AutoCAD VB programmer).

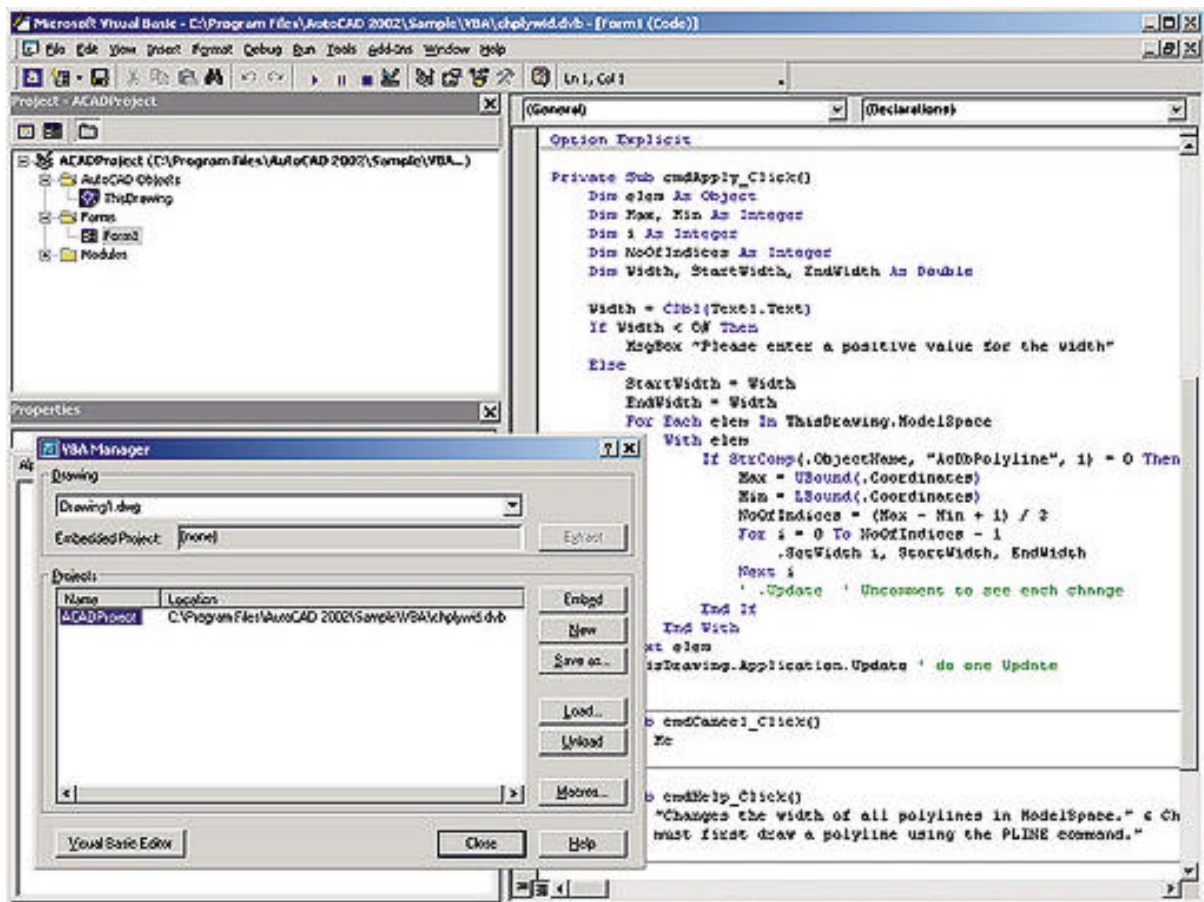


Fig.2: VBA Manager

VBA projects are stored as DVB files, which are accessed from the VBA Manager in AutoCAD (VBAMAN command). Once a DVB file is selected, it can be edited within the Visual Basic editing window to create the required objects, forms, modules, and macros that comprise the project, as shown in Figure 2.

Afterwards, the DVB file can be loaded into AutoCAD, and the new macro can be executed within AutoCAD. Macros can be run from within the Visual Basic Editor (note the RUN pop-down in Figure 2) or from AutoCAD's command line using the VBARUN or -VBARUN commands.

VBA projects are portable because they are contained within a single DVB file. It's easy to email a DVB project to other CAD managers and allow them to load it for access to any macros contained in the project file.

The use of VBA as a customization tool is a very typical path within the applications developed under the Windows platform. This development environment is now under the umbrella of the .NET initiative, which establishes a new model of development and execution.

2.2 MicroStation

MicroStation, from Bentley Systems, is another well known CAD platform to consider when reviewing other approaches to the problem of providing customization tools. Microstation V8 is actually the third generation of MicroStation. The most relevant features of this new version are:

- Transaction-based CAD
- A new DGN format

- Bentley's adoption of VBA
- Support for Microsoft .NET

MicroStation V8 signals the full return of Bentley to the Microsoft fold. While V8 still supports applications written in C, C++ and Java, the Company has dropped Java in favor of VBA and C#. With these APIs, the entire functionality of the software is exposed to application developers. Then, Bentley only supports Windows operating systems. Bentley Systems will also build on top of Microsoft .NET future versions of MicroStation.

2.3 Conclusions of the analysis

General purpose CAD Systems provide for a very powerful set of tools to build specific applications on top of the basic platform. The target audience for these tools is a well established software industry devoted to develop specific sector applications, which are distributed and sold as add-on programs to the general platform.

This fact highlights the most important difference with respect to the type of tools which are targeted in this paper. The requirements of a development environment for a non programmer user (a shipbuilding designer) are much more restrictive than when the target audience are software developers.

The user friendliness of the environment, the simplicity of the programming language and the facilities for executing and debugging the procedures are critical factors in order to provide the designer with a feasible tool to develop his/her own procedures.

An emerging trend in many companies developing CAD Systems is to fall entirely into the Microsoft world. This strategic decision is enforced by the advent of the .NET era.. .NET is a powerful platform which comes with a common runtime for different languages, includes JIT compilation, a powerful remoting infrastructure (including web services) and provides for one of the smartest component model available in the market.

But to develop software in the .NET environment is a skill which is far above the capabilities which are required for a seasoned ship designer. To point to the correct solution, a much simpler environment must be provided.

Another relevant factor is that a shipbuilding CAD System is dealing with a complex product model with topological relations and elaborated business rules. Uncontrolled access to this model could leave it in an inconsistent state. For this reason, the targeted environment should be used as a controlled platform to access the product model data which guarantees the consistency of the performed operations.

Taking into account all these factors, the following paragraphs will describe the future scenario for this new functionality provided by the new generation of shipbuilding CAD Systems.

3. Designer Development Solution

3.1. The environment

The development environment which is suitable for a designer to create their own procedures should be quite different from a typical software development environment. The development process for an end user must be simple and as close as possible to the normal use of the CAD System.

This condition leads to the fact that the CAD System should be the development environment for the designer. In this way, the development environment must be contemplated as an additional functionality provided by the CAD System. The development environment must provide for a user

friendly editor in which to write the procedures. Once a new procedure is written, the execution environment should be again the CAD System.

Then, the CAD System must also supply the runtime environment for the procedures developed by the user. The most effective way to provide for a specific runtime environment is by means of a virtual machine or a command interpreter embedded into the CAD System. Consequently, the procedural language used to develop customer procedures should be any of the several well established scripting languages that exist.

When creating procedures with the scripting language provided by the CAD System, the designer must have the possibility to invoke existing commands and to define interactive dialogs to get the data supplied to the commands.

This environment is at least as controlled as the normal usage of the application. But this should be a very restrictive environment. In order to program most sophisticated actions and to create really new commands, the designer should have access to a more finely grained set of functions accessing the product model in a controlled way. This set of functions constitute a designer development API which must guarantee the consistency of the accessed product model.

Finally, a customization environment must supply some configuration tools in order to add new commands to the menu, to record macros on the fly within an execution path in order to edit it later to add interactivity to the macro and other similar features to facilitate the automation of repetitive tasks by the designer.

3.2. Scripting languages

Considering the environment in which the scripting language has to work, it must play a role of intermediary between the core application and the end user. The scripting language itself must be powerful enough to deal with the programming language in which the CAD application has been developed and simple enough to be managed by the end user.

The best choice should be an object oriented scripting language but not as strongly typed as are C++, C# or Java, which are the typical languages to develop CAD applications. Strongly typed languages are more appropriate for programs being compiled. For scripting programming, strongly typed languages produce a much more verbose coding, which should be difficult to follow and manage by end users. They are also much more difficult for development.

The search for an object oriented (but not strongly typed) scripting language reduces the myriad of scripting languages available to two choices: Python and ECMAScript (the standardized version of JavaScript).

4. The proposed solution

4.1. The FORAN System

The proposed solution is based on the FORAN System, which is one of the most important CAD Systems specialized in shipbuilding. Several features distinguish the FORAN System from others shipbuilding CAD applications:

- The use of a dual geometric model (surface and solid)
- The surface model based on NURBS
- The highest topological definition
- The storage of the complete product model into a relational database
- The most comprehensive coverage of the design process

To these outstanding features, it has been added recently a complete development environment for designers. This environment has been created taking into account all the previous considerations.

The FORAN System development language is C++. One of the most important components of the FORAN System is Qt from TrollTech.. Qt is a C++ toolkit that supports the development of multi platform GUI applications with its “write once, compile anywhere“ approach. Qt has excellent support for many programming domains: 2D and 3D graphics, internationalization, XML, etc.

Another new product offered by TrollTech is QSA , which stands for Qt Script for Applications. QSA is a cross-platform toolkit for making C++ applications scriptable using an interpreted scripting language, Qt Script. Qt Script is based on the ECMAScript standard. Microsoft's JScript, and Netscape's JavaScript are also based on the ECMAScript standard.

QSA always works with one current scripting project that contains all the forms and files in which all the functions and classes are implemented. To facilitate the development process, it is provided the QSA Workbench, a development environment for editing, managing, and running scripts within each project. QSA Workbench includes an output window to view errors that occur when running the script.

Built on top of the FORAN System and using QSA as scripting infrastructure, the FORAN Development Environment (FDE) provides for all the capabilities of QSA complemented with the corresponding API of functions and classes to access the FORAN System product model in a controlled way.

The most relevant features of FDE are:

- It is an easy to learn full featured object oriented language (ECMAScript) which permits implementation of classes and functions.
- It is interpreted, no compilation/linking necessary. No customer adapted program versions are needed.
- Specially suited for writing customer tailored commands and functions in a fast and flexible way.
- Unicode support.
- Simple dialogs may be made for user input, future versions will support creation of any complex dialog by use of a graphical tool (Qt Designer).
- FDS (FORAN Document System) documents may be created with FDE. The documents may be formatted and contain tables, graphics etc.
- File output is supported which makes it easy to generate files that may be used as input for other applications. Support for writing XML DOM documents.
- Support for process handling (sub-processes).
- Any existing application command may be launched and data may be transferred to the dialogs.
- The scene may be traversed and the objects are accessible for fetching information or manipulation.
- Command sequences may be recorded as FDE programs for later to be executed or modified.
- Database access API available.
- A comprehensive online and printed reference manual with cross references is available.
- New commands implemented with FDE may be added to the user configurable custom-menu.

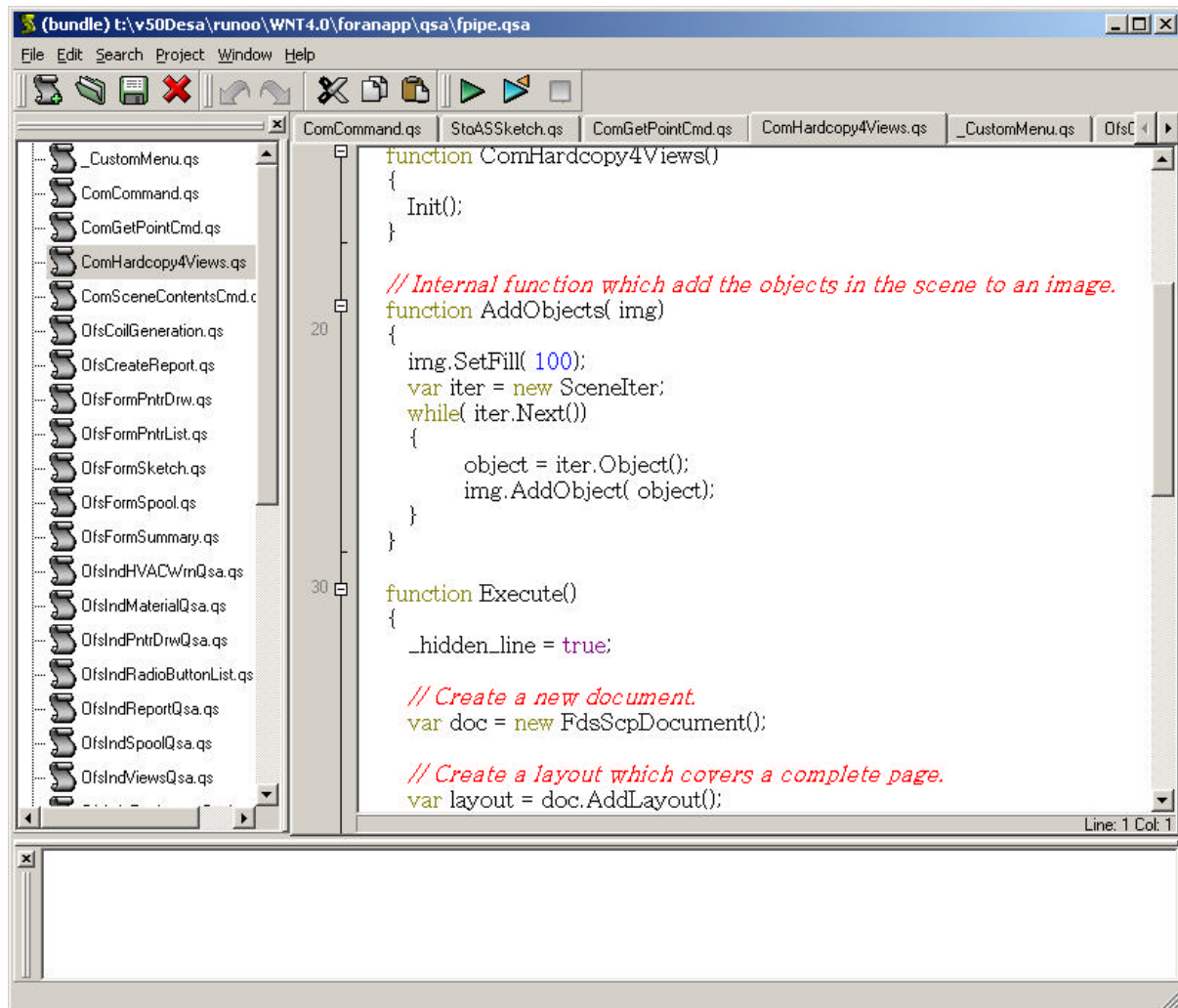


Fig.3: QSA Workbench

4.2. Benefits

There are many benefits that can be obtained from a development environment as it has been described in the previous paragraphs. An advanced user of the application, with a short training and having available the complete documentation, should be able to perform the following tasks:

- Extraction of data from the product model.
- Tailor-made reports with additional graphics.
- Implementation of customer specific tasks.
- Implementation of interactive commands.
- Automation of repetitive tasks for designers.
- Integration of the FORAN System with other systems

In the following paragraphs, these benefits are explained more in detail, with some examples and organized by the following concepts:

- Automation
- Customization
- Integration

4.2.1. Automation

The objective of automation is to reduce drastically the elapsed time of the design process. In this context, automation means the capability of programming action sequences which are repeated over and over within the same design process or between different design processes.

The automation concept targets two main problems which arise in the design process:

- Repetitive tasks, which are straightforward to follow, but are very time consuming
- Complex problems which appear from time to time and which require experience and skills from the designer

Creating a library of scripts which are able to perform repetitive tasks and to solve complex design problems is an asset for the shipyard and / or the design office. This library can be interpreted as a media to store the most valuable knowledge of the company. In this way, the CAD System provided with the user development environment is acting as a knowledge management tool in its most pure conception.

Following there are some examples of automation created with the proposed solution.

Macro recording example

Any user may automate his work by using macro recording. This consists of recording the commands executed, together with the corresponding inputs introduced in the dialogs.

The user just starts the recorder and execute the command sequence that he wants to save as a macro.

After terminating the recorder the source code generated may be used as a basis to implement more sophisticated macros with user interaction, loops etc.

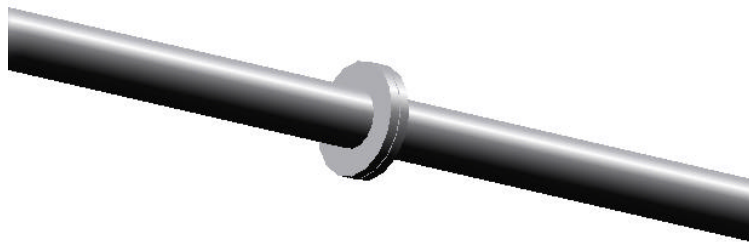


Fig.4: Example: inserting flanges in a pipe-line.

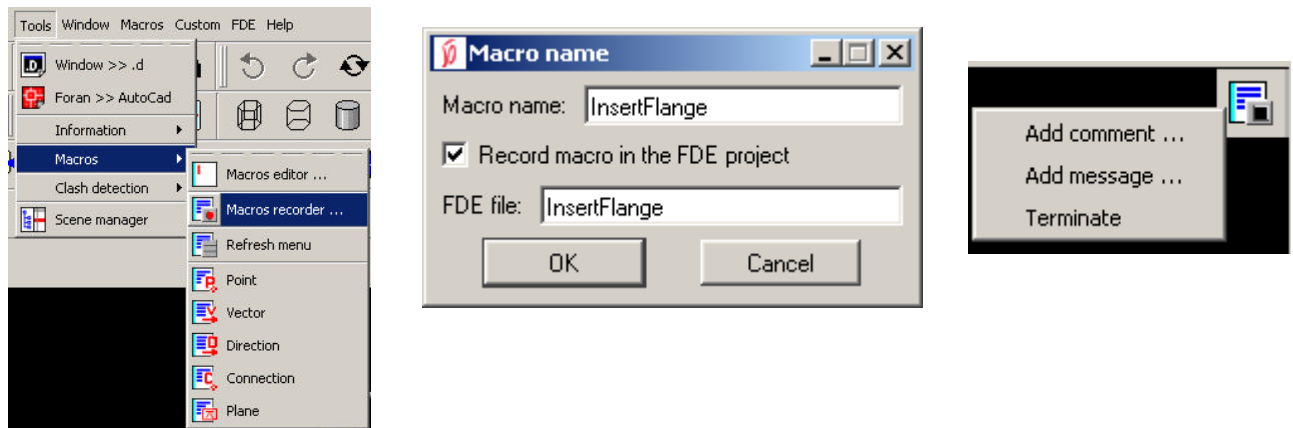


Fig.5: Example: inserting flanges in a pipe-line. Registering the macro.

The result of the macro recorder is stored in the FDE project and is available in the Workbench for editing and direct execution. The user may use the macro recorded as a template for a more sophisticated macro applying specific design rules.

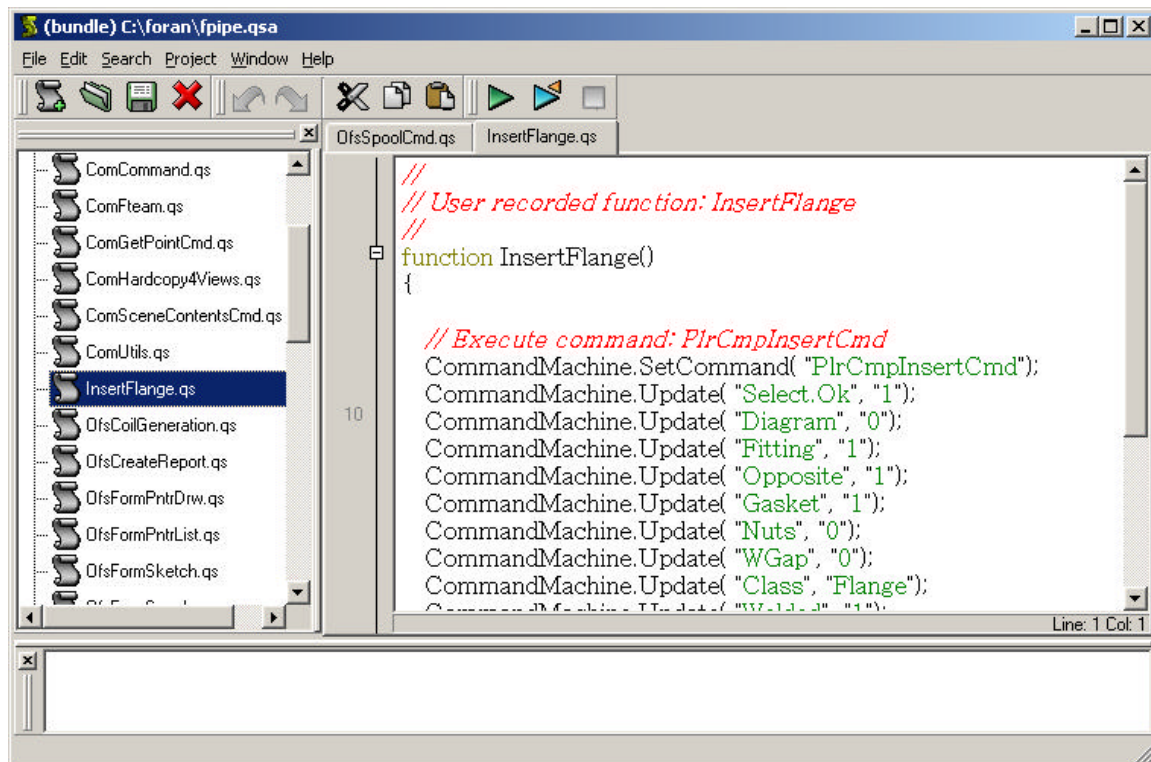


Fig.6: Example: inserting flanges in a pipe-line. Code generated automatically, which can be modified by the user.

Coil routing example

In this example, an user develops a procedure to perform a sequence of pipe routings, following his own rules, to define a coil. The procedure could be made much simpler for creating a specific type of coil (rules for coil with fixed directions, dimensions dependant of the nominal diameter, etc.). The example shows a general coil routing utility. The sequence of actions is:

- Selection of the coil plane
- Distance to the plane selected.
- Start point for the coil.

- Direction of the start of the coil.
- The advance direction of the grid.
- Coil length
- Grid length
- Intermediate pipe length

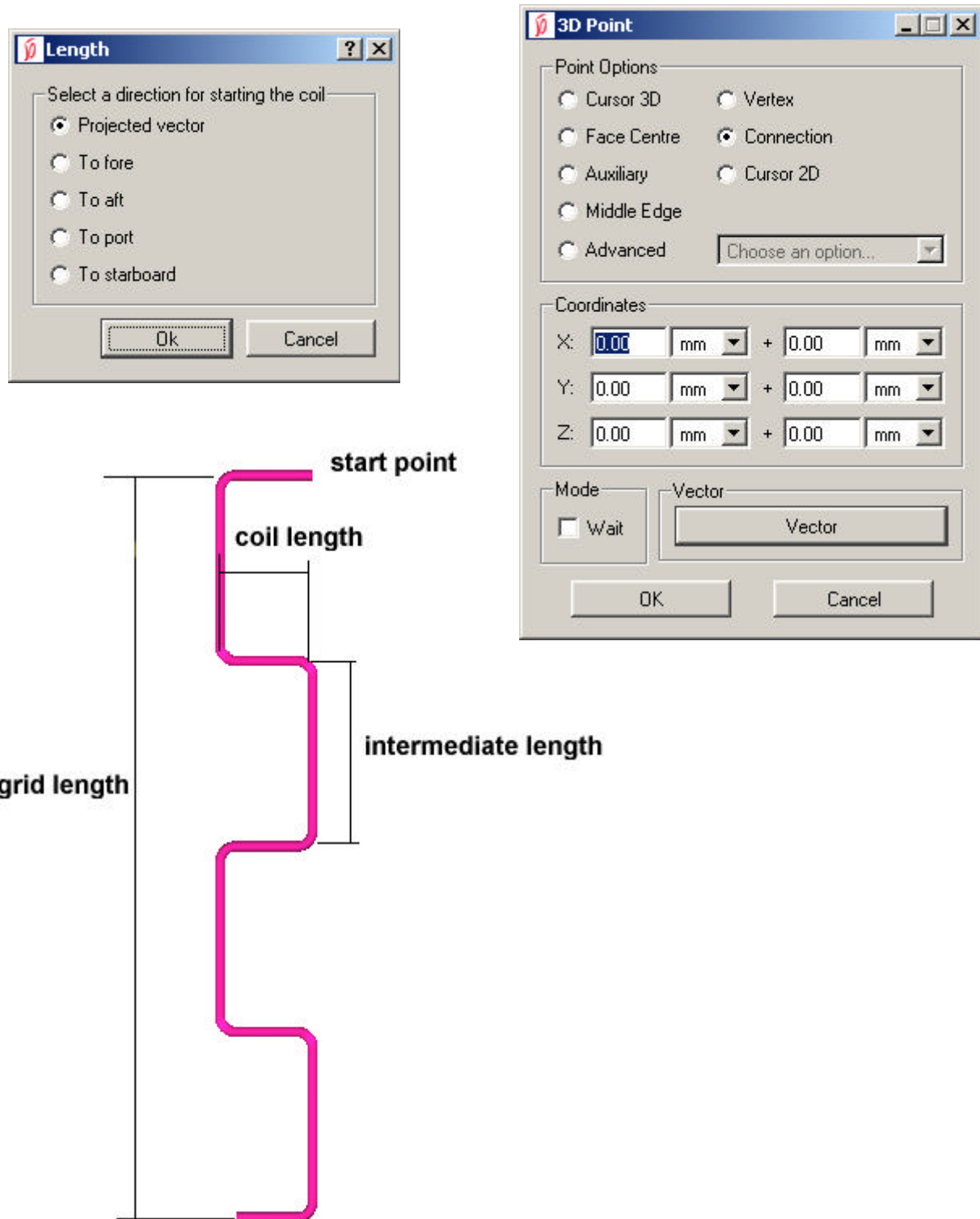


Fig.7: Example. Dialog made with the development environment, the standard 3D point dialog used to obtain the start point for the coil and the resulting pipe coil created

4.2.2. Customization

Customization refers to the process of adapting the standard outputs of a system to those required by a specific customer. In a more general sense, it could include also some of the processes defined in automation. Here, the term is used in a more restrictive way because automation has been already considered.

One of the biggest challenges of customization is the couple of interrelated processes denominated internationalization (I18N) and localization (L10N). Nowadays any commercial application is supposed to provide this functionality “out of the box”, as it happens with the proposed solution. Anyway, the requirement remains for the development environment and again the proposed solution comply with this.

The most important objective which is targeted by the customization process is to avoid unnecessary changes in the communication paths already established between the design office and the workshops. This changes would have a big impact in the internal production processes, having strong effects on the planning and quality of the production.

Customization is obtained with a combination of functions giving access to the product model data and other functions providing tools to manage drawings and reports. In order to illustrate how this problem is approached by the proposed solution, another example is shown.

Spool drawing generation example

This example produces an automatic generated spool drawing which is implemented as a new command using FDE. The complete implementation is done in the scripting language which permits a total flexibility to make changes for the advanced user.

The steps when executing the command are:

- The command opens a dialog to choose which pipes to process (selected pipes or all pipes in the scene), and for introduction of some additional information.

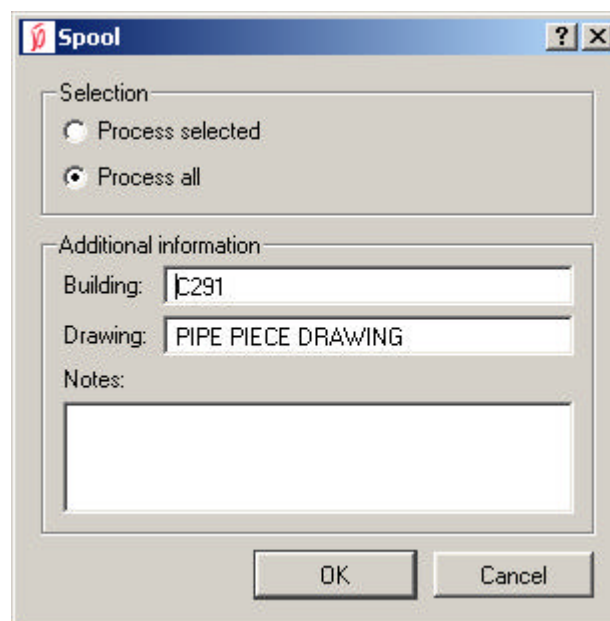


Fig.8: dialog programmed by the user to select pipes for the output

- The FDE program processes the pipes and generate the graphics and material list. When the pipe is in one plane only one projection is created, if the pipe is located in more than one plane four graphical projections are created.
- The report generated is opened in a print pre-viewer, alternative outputs could be sending directly to the printer, or saving the document to file as e.g. a TIF raster file.

[illegible]

Fig.9: Spool drawing generated by the user program

4.2.3. Integration

The set of IT technologies that a company owns has only one objective, to support the business processes of the company. In order to reach this objective, the different software components that participate in the business processes of any company have to collaborate. The main purpose of this collaboration is to avoid the input of the same data several times, not only to save time, but mainly to avoid errors.

Integration is the process of getting different software components to collaborate. As most of the software components are structured in different tiers, integration can take place at any of these tiers. The most common tier to perform integration is data tier. Integration at the data tier can be reached in different ways.

When the data tier is supported by a relational database, the most common way of implementing data integration is by the use of intermediate tables updated by means of triggers. A more general way of data integration is by importing / exporting neutral files. Today, this leads to the use of XML files, validated by public or proprietary schemas.

Consequently, the development environment should provide for facilities to manage XML parsers and relational database connections, commands and result sets from queries. To illustrate these processes, some examples created with the proposed solution are provided.

Use of XML

With the development environment the advanced user may create tailor-made XML documents for his specific needs. Most of the times, these outputs shall be used as inputs for other applications, using a public schema.

```
// Create a DOM document.
var doc = new XmlDocument( "MyDoc" );

// Add an element to the document.
var elem = doc.AddElement( "MyElement" );
// Add a text node to the element.
var txt = elem.AddTextNode( "My text node" );
// Add a comment to the element.
elem.AddComment( "This is the text node" );

// Add an element as a child of the first element.
var elem2 = elem.AddElement( "TheSecondLevel" );
// Add a text node.
elem2.AddTextNode( "The second level text" );
// Add attributes the the element.
elem2.SetAttribute( "Text", "The text value" );
elem2.SetAttribute( "Integer", 1 );
elem2.SetAttribute( "Float", 1.234 );

// Open a file for output and write the contents.
var ofile = new OFile( "output.xml" );
var xml = doc.toString();
ofile.Write( xml );
```

Database access

The FORAN System uses an Oracle relational database for storing the product information. By allowing access from FDE virtually any information may easily be extracted from the database. Database views are defined in the FORAN system to simplify the access to the information by joining data from several database tables. This example illustrate a simple process of extracting information.

```
// Simple example reading information from the
// database by using a FrnDBView object.
// Print the "oid" and "userid" for all the elements in
// system "GENE".
function PrintUserId()
{
    // Create the view object with a search condition.
    var view = new FrnDBView( "v_element",
                             "syst_userid='GENE'");

    // Iterate over all the records in the view.
    while( view.Next())
    {
        print( view.Value( "oid") + " --- " +
               view.Value( "userid"));
    }
}
```

Use of auxiliary tables

Auxiliary database tables may be used to add additional information to the product model. It is also useful for transferring information to other systems directly by storing in the database. It is possible to execute any SQL statement in the FORAN System database and even in an external database.

```
//
// Function which adds an oid with an additional
// description to the table FDE_DESCR
//
function InsertDescr( oid, descr)
{
    // Create the SQL statement text.
    var sql = "INSERT INTO FDE_DESCR (ID,DESCR) " +
              "VALUES (" + oid + "," +
              "'" + descr + "'" );

    // Execute the statement.
    if ( !FrnDBUtils.Execute( sql))
        MessageBox.critical( "SQL failed: " + sql,
                             MessageBox.Ok);
}
```

5. Conclusions

The most important conclusion is that the way in which CAD applications are being used in the shipbuilding environment is going to change drastically in the next years. It is obvious that this statement can be stated at any moment, due to the dynamic nature of the environment in which IT tools are evolving.

This time, the innovative aspect of this evolution is that it could be leaded by the advanced users of the CAD applications. In order to make this scenario possible, the advanced users should be provided with powerful and easy to use development environments, as they have been established in this paper.

The main benefits that can be obtained from this change are extremely important:

- Automation and optimization of the design processes
- Knowledge management of the design skills
- Customization and integration performed by users

Some of the most evolutionary CAD applications, as is the proposed solution, are preparing this scenario. The next years we will see if the corresponding benefits can be converted into a reality.

References

RODRIGUEZ, A.; GONZALEZ, C.; GURREA, I.; SOLANO, L.; *Kernel architecture for the development of cad/cam applications in shipbuilding environments*, 2nd Int. Conf. Computer and IT App. Maritime Ind., COMPIT, Hamburg

GREEN, R. (2004), *AutoCAD Customization Overview*, CADALYST Monday March 15th

CYON RESEARCH CORPORATION; *The newsletter for engineering management. engineering automation report*, November 2001, Volume 11, Issue 11

TROLLTECH; Qt 3.3 White paper, www.trolltech.com

RODRIGUEZ, A.; VIVO, M.; VINACUA, A.(2000), *New tools for hull surface modeling*, 1st Int. Conf. Computer and IT App. Maritime Ind., COMPIT, Potsdam

ALONSO, F.; CEBOLLERO, A.; GOMEZ, A.; RODRÍGUEZ, A.(2002), *Collaborative engineering in shipbuilding*, ICCAS'2002

A Heuristic for the Container Pre-Marshalling Problem

Andreas Bortfeldt, University of Hagen, Hagen/Germany, Andreas.Bortfeldt@FernUni-Hagen.de

Abstract

This contribution is faced with a situation that occurs frequently in a container terminal of a seaport. After a certain set of containers has arrived at the terminal, the containers are stored in a so-called bay. Some hours later the containers are loaded onto a ship and then transported to their destination ports. A bay consists of several stacks. Each of the stacks can accommodate the same number of containers, i.e. the maximal stack height is a constant. Although the container terminal may have a lot of bays, it is assumed for technological reasons that all containers – belonging to the considered set – are stored in the same bay. After arrival, the containers are distributed to the stacks in an arbitrary manner. The resulting initial layout of the stacks is not suitable with regard to the following loading operation: containers that are to be loaded first, are often buried beneath other containers. Therefore, at first a pre-marshalling process is performed to get an objective layout that corresponds to a given loading sequence of the containers. The pre-marshalling is carried out in a number of steps; in each step a container is moved from the top of a certain stack to the top of another stack by means of a gantry crane. Of course, the number of steps should be as small as possible. The container pre-marshalling problem (CPMP) may now be formulated as follows: for a given initial layout and a given objective layout determine a sequence of container movements with a minimum number of steps. The contribution proposes a branch-and-bound approach for solving the CPMP. A set of benchmark instances is introduced with bay sizes (no. of stacks, stack height) corresponding to those used in practice. The numerical results are promising: for many instances near-optimal or even optimal solutions have been computed within short computation times.

1. Introduction

This paper deals with the so-called container pre-marshalling problem (CPMP), which occurs in connection with loading container ships with export containers in a seaport container terminal. The CPMP demands that a layout that matches the ship's planned loading sequence is drawn up for the containers for a defined ship with the least effort in the terminal yard. Only movements of containers within the same bay are permitted. A heuristic for a main variant of the CPMP is proposed with which even large problem instances can be solved with a high solution quality and with little time required.

2. Description of the container pre-marshalling problem

Export containers delivered on shore are first of all transported to the yard of the container terminal. They usually spend several days here before being transported to the berth of the planned container ship and loaded on the ship there. The yard is divided into several blocks, and each block consists of several bays. Each bay in a block includes the same number of stacks, Fig.1. Each stack has the same number of slots, i.e. can take the same number of containers. The containers are transported to and from the yard by yard trucks. A gantry crane, e.g. a rail-mounted gantry crane, is available for moving containers in and out of a stack. This crane can only transport one container at a time and only access the top container in a stack. If, therefore, a container lower down in the stack is to be taken out of store, all the containers in the slots above it have first to be transferred on other stacks. A movement of a container from one stack to another stack is referred to as re-handling. Fig.2 shows a bay with a gantry crane.

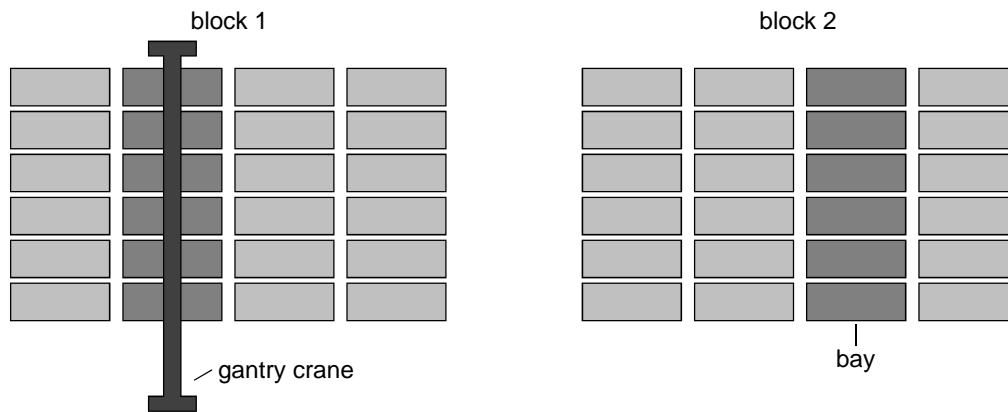


Fig.1: Container terminal store with blocks and a gantry crane (view from above)

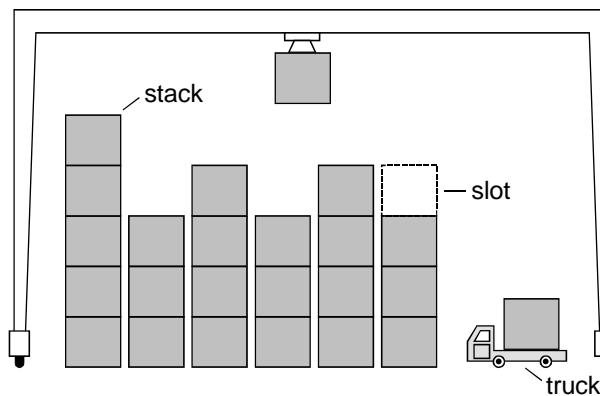


Fig.2: Bay with gantry crane

A ship is loaded with the help of a stowage plan. In order to draw up a stowage plan the containers that are to be loaded are usually divided into groups in accordance with the following attributes: container type or length (20', 40',...), destination port and weight class (light, medium, heavy). All containers that have the same attributes belong to a container group. A container group, e.g. the group (20', Hamburg, light), is assigned in the stowage plan to each slot in the ship that is to be loaded. This means that any container of the planned group is to be allocated to the relevant slot.

To some degree, the stowage plan determines the sequence in which the containers are to be loaded onto the ship. For reasons of stability, heavier containers are loaded in lower positions, lighter containers in higher positions in the ship, so that heavier containers in general are loaded before lighter ones. The ports of destination also influence the loading sequence, because, in the interests of reducing transshipping expenses in a port of destination, where possible, containers for closer ports are not placed underneath containers for more distant ports.

On the whole it is assumed here that the following applies for the loading sequence derived from the stowage plan: the containers for a defined ship are loaded in groups. With a suitable indication of the groups in a stowage plan from 1 to G , all containers in a group g must be loaded first, before containers in the group $g + 1$ can be loaded ($g = 1, \dots, G - 1$).

Loading a ship can only be handled efficiently if no, or few, re-handlings have to be carried out in the yard during loading, because considerable delays occur otherwise. Let it be assumed that only containers for a single ship are stored in a yard stack. The result from the loading sequence is that the following general stacking condition (GSC) should already be fulfilled at the start of loading: none of the stacks that are reserved for a ship that is currently to be loaded holds a container in group g underneath a container in a group $g' > g$ ($1 = g, g' = G$). In other words it must not be found that a container that is to be loaded early is underneath containers that are to be loaded later.

When they arrive in the terminal yard the containers designated for a particular ship are not usually stacked in accordance with the GSC. There are several reasons for this. Stacking containers in homogenous groups, in which the GSC is taken into account from the start, often takes up too much space in the yard. In addition, the information that is required for allocation to a group (port of destination, weight class) is often not available when the containers are taken into storage, or is faulty, or the port of destination is subsequently changed.

In order to be able to guarantee a container layout in accordance with the GSC when loading the ship starts, the necessary re-handlings are often performed in the time remaining before loading (usually a few hours). This process is referred to as pre-marshalling the containers.

The following assumptions are made with regard to the pre-marshalling:

- (1) Pre-marshalling is carried out for one ship only. It is therefore assumed as well that one or more complete bays are reserved for a ship.
- (2) The time that is required for moving a container from one stack to another stack in the same bay is not dependent on the distance of the stacks. This assumption is justified because most of the time necessary for a re-handling is used by the gantry crane for picking up and depositing the containers.
- (3) Only re-handlings within a bay are permitted. The justification for this assumption is that re-handling over the borders of a bay and using a gantry crane is very time-consuming. Therefore, if the containers for a ship are located in several bays, pre-marshalling is carried out separately for each bay.

Any filling of a bay with containers for a ship is referred to below as a layout. Pre-marshalling starts with the initial layout that is found after the containers are brought into the yard. An unspecified final layout is any layout that fulfils the GSC. In contrast, a specified final layout fulfils the GSC and, at the same time, stipulates the container group for each occupied slot. The pre-marshalling should take place with the least possible effort. The result of this is the container pre-marshalling problem, which occurs in two variants:

- (1) CPMP with unspecified final layout:
Determine a sequence of the fewest possible re-handlings that transforms an initial layout into any unspecified final layout.
- (2) CPMP with specified final layout:
Determine a sequence of the fewest possible re-handlings that transforms an initial layout into a given specified final layout.

This paper only deals with the CPMP with an unspecified final layout.

Until now there have been very few articles on the CPMP. Steenken et al. (2004) describe the essential logistical processes in container terminals and provide an overview of the literature on this subject. There is a comprehensive representation of the decision problems in container terminals from Vis and de Koster (2003). Chen (1999) examines the storage operations in container terminals and the strategies that are used here. Kim (1997) develops a method on estimating the number of re-handlings for import containers. Kim and Bae (1998) deal with the rearrangement of containers in the yard to accelerate the subsequent loading process. However, while the movement of transfer cranes between bays is taken into account, there is no detailed examination of the rearrangement of containers within a bay. This also applies to other papers, e.g. for Kim and Kim (1998), Kim and Kim (1999) and Kim et al. (2000), which consider the transport of containers within the yard. Lee and Hsu (2002) and Lee and Hsu (2003) develop a mathematical model for the two variants of the CPMP described above. On the basis of this model less complex CPMP instances can be solved to optimality with the help of standard optimizing software.

3. Problem formulation

The CPMP with an unspecified final layout is expressed formally below. In addition, the terminology and notation required for drafting the heuristic will be introduced. The stacks of a bay are numbered in accordance with $s = 1, \dots, S$ ($S > 1$) and the slots of a stack are indexed from bottom to top in accordance with $h = 1, \dots, H$ ($H > 1$). The containers located in the bay during pre-marshalling are divided into G groups ($G > 1$). The indexing of the groups in accordance with $g = 1, \dots, G$ corresponds to the loading sequence, as has already been explained. For each group g there are $C(g)$ ($C(g) \geq 1$) containers, the total number of containers is C .

A layout of the bay is represented through a matrix L with H lines and S columns. The matrix L assigns a group index g ($0 = g = G$) to each slot, given by a pair (h, s) ($h = 1, \dots, H, s = 1, \dots, S$). An entry $L(h, s) = 0$ means that the slot (h, s) is empty, while an entry $L(h, s) = g > 0$ indicates that the slot (h, s) is filled by a container of group g . Only layouts L are permitted that fulfil the following conditions:

- If $L(h, s) = 0$, then $L(h', s) = 0$ applies as well for all $h' = h + 1, \dots, H$ ($h = 1, \dots, H-1, s = 1, \dots, S$).
- The number of matrix elements $L(h, s)$ with the value g equals $C(g)$ ($g = 1, \dots, G$).

Any layout L is deemed to be an (unspecified) final layout, if the condition $L(h + 1, s) \geq L(h, s)$ is fulfilled for all $h = 1, \dots, H-1$ and all $s = 1, \dots, S$.

A re-handling of a container is called a move and represented by a pair (d, r) of different stacks ($1 = d, r = S, d \neq r$). The first stack (d) is designated the donator, the second stack (r) the receiver. For a given layout L a move $m = (d, r)$ is deemed to be applicable (or permitted) if the following conditions are fulfilled:

- $L(1, d) > 0$, i.e. the donator is not empty.
- $L(H, r) = 0$, i.e. the receiver is not full.

If a permitted move $m = (d, r)$ is applied to a layout L , the following layout L' results in accordance with the stipulation:

- $L'(top(L, d), d) = 0$.
- $L'(top(L, r) + 1, r) = L(top(L, d), d)$.
- $L'(h, s) = L(h, s), h = 1, \dots, H, s = 1, \dots, S, (h, s) \neq (top(L, d), d), (h, s) \neq (top(L, r) + 1, r)$.

Here $top(L, s) = \max\{h \mid L(h, s) > 0, h = 1, \dots, H\}$, $s = 1, \dots, S$. A move therefore displaces the highest container of the donator to the lowest free slot in the receiver.

A move sequence (m_1, m_2, \dots, m_n) ($n \geq 1$) is permitted with reference to a layout L , if the move m_1 is applicable to L and each move m_{i+1} is applicable to the layout resulting after m_i ($i = 1, \dots, n-1$). The CPMP with an unspecified final layout can be formulated as follows: determine a move sequence s with n moves for a given initial layout L_{init} that meets the following requirements:

- s can be applied to L_{init} .
- The layout resulting from the n -th move is a final layout.
- If s' is any move sequence with n' moves that fulfils (i) and (ii), then $n' \geq n$.

The following stipulations are made, whereby L is any layout:

- Let $L(h, s) = g > 0$. The container g is called badly placed, if $h > 1$ and $L(h-1, s) < g$, or if there is a badly placed container in a slot (h', s) , $1 < h' < h$. Otherwise the container g (in slot (h, s)) is well placed.
- $nC_{bad}(L)$ designates the number of all badly placed containers in L and $nC_{bad}(L, s)$ designates the number of all badly placed containers in stack s of L ($s = 1, \dots, S$). Stack s is called clean if $nC_{bad}(L, s) = 0$, otherwise it is dirty.

- (3) The number of all badly placed containers in group g in L is designated as demand $dem(g)$ of group g ($g = 1, \dots, G$). The cumulative demand $cdem(g)$ of group g results as the sum of demands of groups $g' = g, \dots, G$ ($g = 1, \dots, G$).
- (4) Every slot (h,s) that fulfils the following conditions is seen as a potential supply slot for group g in layout L ($h = 1, \dots, H, s = 1, \dots, S, g = 1, \dots, G$):
 - Stack s is not empty and the highest well placed container in s belongs to group g .
 - The slot (h,s) is above the highest well placed container in s .
 The number of all potential supply slots is defined as the potential supply $psup(g)$ for the group g . The cumulative potential supply $cpsup(g)$ for the group g is stipulated analogously to the cumulative demand. The cumulative demand surplus $csdem(g)$ of group g is the difference $cdem(g) - cpsup(g)$ ($g = 1, \dots, G$).
- (5) A current supply slot for the group g is a potential supply slot for g that is in a clean stack. In addition, an empty stack provides H current supply slots for group G . The current supply $csup(g)$ for the group g in the layout L is the number of current supply slots for g ($g = 1, \dots, G$). The current supply of a layout L_1 is deemed to be greater than the current supply of a layout L_2 if there is a group g for which $csup_1(g) > csup_2(g)$, while $csup_1(g') = csup_2(g')$ for all $g' = g + 1, \dots, G$.
- (6) Let the move m be applicable to layout L . The move m is from the type Bad-Good (in short BG), if the moved container was badly placed before the move and is well placed after the move. The move types Bad-Bad (BB), Good-Good (GG) and Good-Bad (GB) are defined analogously. A BX move is either a BG move or a BB move. GX moves are stipulated analogously.

4. Design of a branch-and-bound procedure

The branch-and-bound procedure is designed in three steps. First of all, a lower bound is derived for the number of moves for transforming an initial layout into a final layout. After this, the approach of the tree search is presented, which is based on the concept of compound moves. Finally, the generation of compound moves for a layout is described.

4.1. A lower bound for the number of moves

A lower bound for the number of moves that are necessary for transforming any layout L into an (unspecified) final layout is required in the branch-and-bound procedure for reducing the search effort. Applied to the initial layout L_{init} of a problem instance the lower bound also serves the evaluation of the achieved solution quality when the procedure is tested.

In order to acquire a lower bound nM' for the number of required moves with regard to a given layout L , lower bounds nM_{BX}' and nM_{GX}' are derived separately for the move types BX and GX , respectively. The lower bound nM' for the total number of moves then results in accordance with:

$$nM' = nM_{BX}' + nM_{GX}'. \quad (1)$$

a) A lower bound for the number of BX moves

At least one Bad-Good move must be carried out for every badly placed container in a given layout in order to reach a final layout. This means that a total of at least $nC_{bad}(L)$ BG moves must be performed. If all stacks in layout L are dirty, BG moves can only be carried out after all badly placed containers have been removed in at least one stack. This can obviously only be done through Bad-Bad moves, since there are still no slots available for the badly placed containers on which they would be well placed.

The consequence is that (in the beginning) at least $\min\{nC_{bad}(L,s) \mid s = 1, \dots, S\}$ BB moves must be carried out. If layout L contains at least one clean stack, the required minimum number of BB moves

naturally equals zero. Altogether, the following lower bound results for the number of BX moves on the transformation of a layout L into a final layout:

$$nM_{BX}' = nC_{bad}(L) + \min\{nC_{bad}(L,s) \mid s = 1, \dots, S\}. \quad (2)$$

b) A lower bound for the number of GX moves

The determination of a lower bound for the number of GX moves on the transformation of a layout L into a final layout is shown with the example of the layout in Fig.3 with $S = 6$, $H = 4$, $G = 4$, in which the well placed containers are shown darker.

Height						
:						
4	1					3
3	4				3	3
2	2	4	3	2	4	3
1	2	3	1	1	1	1
Stack:	1	2	3	4	5	6

Fig.3: Example of a layout

The minimum number of GX moves is determined as follows for the example layout L :

- (1) First of all, the cumulative demand, the cumulative potential supply and the cumulative demand surplus are determined for each container group (cf. Section 1, (3) to (5)). The values listed in Table I result for layout L .

Table I: Demand and supply for container groups in the example layout

Container-group g	Demand	Potential supply	Cumulative demand	Cumulative pot. supply	Cum. demand surplus
4	3	0	3	0	3
3	5	3	8	3	5
2	1	2	9	5	4
1	1	12	10	17	-7

- (2) In the given case the maximum cumulative demand surplus is 5 and occurs in container group 3. This obviously means that if all well placed containers remain in their slots, there are still 5 slots missing for placing the badly placed containers in groups 3 and 4 in good slots. Because the stack height $H = 4$, well placed containers must be removed in at least $nS_{GX} = 2$ stacks, i.e. GX moves must be carried out from 2 stacks to provide the missing 5 slots.
- (3) In stack 2 no additional slots can be created in GX moves for containers in groups 3 and 4. For the slots above container 3 were already considered in the potential supply and removing this container does not create any additional space, since it has to be put in a good slot again itself.
- (4) In the other stacks at least one well placed container must always be removed to create additional slots for good placing of the containers in groups 3 and 4. This means that a total of at least 2 GX moves are necessary, or $nM_{GX}' = 2$ is a lower bound for the number of required GX moves.

If empty stacks are available, in contrast to the example layout L , the number nS_{GX} of those stacks in which GX moves are necessary must be reduced by the number of empty stacks nS_{empty} . If nS_{GX} is then

still greater than zero, the minimum number of GX moves must be determined taking the stacks that are not empty into account. Otherwise $nM_{GX}' = 0$.

In general the lower bound nM_{GX}' explained in the example can be defined as follows. Let g^* be the group with the maximum cumulative demand surplus $csdem(g^*)$. In addition, let nS_{GX} be defined in accordance with $nS_{GX} = \max(0, \lceil csdem(g^*) / H \rceil - nS_{empty})$. Let $nC(g^*, s)$ designate the number of well placed containers in groups $g < g^*$ in stack s ($s = 1, \dots, S$). Let the stacks be renumbered so that:

- There are no empty stacks among the first nS_{GX} stacks and the highest well placed container belongs to a group $g < g^*$;
- $nC(g^*, s) = nC(g^*, s + 1)$ for $s = 1, \dots, nS_{GX} - 1$.

The lower bound nM_{GX}' is then determined in accordance with :

$$nM_{GX}' = \sum_{s=1}^{nS_{GX}} nC(g^*, s). \quad (3)$$

The case that $csdem(g^*) = 0$ or $nS_{GX} = 0$ is also included: in accordance with Eq.(3) $nM_{GX}' = 0$.

4.2. Tree search approach

A tree search is carried out to determine a move sequence of minimal length that transforms an initial layout into a final layout. The nodes of the tree represent layouts; the root node in particular corresponds to the initial layout and each leaf node corresponds to a final layout.

In the tree search a direct successor L' of a layout or node L is generated by means of a so-called compound move. A compound move is a permitted move sequence with regard to L (cf. Section 3). A complete or incomplete solution of a CPMP instance is represented within the tree search through a sequence of compound moves. The number of all moves of s , designated by $nM(s)$, results simply as the sum of all move numbers of all compound moves of s . The generation of suitable compound moves for a layout L will be dealt with later.

There are several measures that serve to keep the search effort within acceptable limits:

- (1) The number of successors L' of a layout L is restricted. At most $nSucc$ ($nSucc > 1$, fixed) different compound moves are applied alternatively to layout L . If L was achieved through the partial solution s , only $nSucc$ continuations of s are taken into account.
- (2) Let s^* be the previously best (complete) solution, s a partial solution and L the appropriate layout. If $nM(s^*) = nM(s) + nM'(L)$ holds, then s is no longer continued, since no new best solution can be achieved from s . The restriction of the tree search with the help of the lower bound $nM'(L)$ identifies the procedure as a branch-and-bound procedure.
- (3) The search is aborted if a time limit is exceeded or if an optimum solution was found, i.e. if the best found solution s^* fulfils the condition $nM(s^*) = nM'(L_{init})$.

The tree search is carried out by means of the recursive procedure `perform_compound_moves`, which is shown in Fig.4. The following initialisings are carried out before the procedure is called for the first time: current solution $s := \emptyset$, best solution $s^* := \emptyset$, current layout $L := \text{initial layout } L_{init}$. At the end of the search the best solution s^* and the appropriate final layout L^* are returned.

```

perform_compound_moves (s {current solution}, L {respective layout})

{abort the search where appropriate}
if  $s^* \neq \emptyset$  and  $nM(s^*) = nM'(L_{init})$  or time limit exceeded then
    return;
endif;

{abort search path if a leaf node was reached; update best solution if necessary}
if L is a final layout then
    if  $s^* = \emptyset$  or  $nM(s) < nM(s^*)$  then
         $s^* := s$ ;  $L^* := L$ ;
    endif;
    return;
endif;

{abort search path if no further best solution can be reached }
if  $s^* \neq \emptyset$  and  $nM(s^*) = nM(s) + nM'(L)$  then
    return;
endif;

determine at most nSucc compound moves  $Cm(i)$ ,  $i = 1, \dots, nCm$ ;

{perform the compound moves alternatively}
for  $i := 1$  to  $nCm$  do
     $s' := s \cup Cm(i)$ ;
    determine layout  $L'$  by applying  $Cm(i)$  to  $L$ ;
    perform_compound_moves ( $s'$ ,  $L'$ );
endfor;
end.

```

Fig.4: Recursive procedure perform_compound_moves

4.3. Generating compound moves

Moves of the Bad-Good type reduce the number of badly placed containers and therefore the distance to a final layout, whereas this does not apply for the other move types. In order to reach a final layout from a layout L with the fewest moves, in particular the fewest moves of types BB , GG and GB must be carried out. This makes the following approach for the generation of compound moves obvious:

- Only two types of compound moves are taken into account. Compound moves of the type 'normal' contain only moves of type BG , while compound moves of type 'extra' contain only moves of the types BB , GG and GB .
- Normal compound moves are used preferably. If at least one BG move can be applied to a given layout L , only normal compound moves are generated; otherwise extra compound moves are generated.

Compound moves are determined in accordance with heuristic rules, which are shown in an overview and separately for the types normal and extra.

a) Compound moves of the normal type

- For each dirty stack s exactly one compound move is generated in which all moves contain the donator s (compound moves with fixed donator). In addition, a compound move is generated in which the donators of the moves can differ (compound moves with mixed donators). Normal compound moves with fixed donator (also) serve the aim of converting the donator into a clean stack, to enable further BG moves. Compound moves with mixed donators should bundle as many BG moves as possible in order to reduce the distance to a final layout as much as possible.
- Only maximum compound moves are generated, i.e. a normal compound move only ends when a BG move (where necessary from the same stack) is no longer possible.
- Compound moves are generated move for move. The next move is determined in accordance with the following criteria:
 - (1) If the donator is not fixed, a move is selected whose container belongs preferably to a high group.
 - (2) If there is any leeway, the receiver is selected in such a way that the difference of the groups of the moved container and of the previously topmost container in the receiver is as small as possible.

Both criteria contribute to ensuring that after a move as many other BG moves as possible can be carried out.
- Finally the generated compound moves are sorted in descending order in accordance with the number of moves and – in case of ties – in descending order in accordance with the current supply of the layout that results from the compound move. The last excess compound moves are cancelled where applicable.

b) Compound moves of the extra type

- In general, for each stack s several compound moves with a fixed donator are generated, whereas no compound move with mixed donators is set. That reflects the fact that the only productive function of extra compound moves consists of enabling BG moves again.
- An extra compound move to the donator s is maximum, if s is empty after the last move. Along with maximum compound moves, all move sequences from the first n ($n = 1, 2, \dots$) moves of a maximum compound move are also seen as complete compound moves. This conforms to the objective of managing with as few moves of types BB , GG , GB as possible.
- Since the donator is established with extra compound moves, only criteria for the selection of the receiver are necessary for determining the next move:
 - (1) The receiver is preferably selected so that a GG move is generated, since this does not lead to a further move for the moved container.
 - (2) If a GG move is not possible, a dirty stack is selected preferably as the receiver, since otherwise further BG moves can be obstructed.
- Finally the generated compound moves are sorted in ascending order in accordance with the number of moves and – in case of ties – in descending order in accordance with the current supply of the layout that results from the compound move. Again, the last excess compound moves are cancelled where applicable.

5. Test of the procedure

The complexity of CPMP instances depends on different factors. With the CPMP variant with an unspecified final layout these are the following factors:

- the dimensions of the bay, in other words the number and height of the stacks,
- the number of container groups,
- the distribution of the containers over the groups,
- the percentage of occupied slots in the bay and
- the percentage of badly placed containers in relation to all containers in the initial layout.

Roughly speaking, the difficulty of the CPMP should increase if one factor increases while the others stay the same. For example, a CPMP instance should be more difficult to solve if the selected dimensions of the bay are larger, but the occupation percentage of the bay and the other factors remain the same. This example also shows that fixing the other remaining factors must be supported: with larger bay dimensions, but sinking occupancy rates (with the number of containers remaining the same, for example), the difficulty of the CPMP will be reduced. With regard to the distribution of the containers over the groups it appears obvious that the difficulty of the problem is then particularly great if roughly the same number of containers accounts to each group.

There are hardly any test examples for the CPMP in the literature. For this reason, a total of 32 problem instances were generated randomly for the CPMP variant that is dealt with here. Along with the above considerations, the data from Lee and Hsu (2003) on the size of bays were observed on the generation of test instances. According to this, in practice bays are regarded as large that cover 16 stacks with height 5. The 32 problem instances are divided into eight test cases each with four instances. The reference data for the test cases are listed in Table II. In all instances the containers are distributed evenly over the container groups. If storage of the containers in the stack is assumed that is random with regard to the general stacking condition (cf. Section 2), the rates of badly placed containers appear to be selected reasonably. On the whole these are obviously complex instances of the CPMP variant that was dealt with. This is also shown by a comparison with an example (the only one) with an unspecified final layout calculated by Lee and Hsu (2003) (cf. Tab. 2, last line). The heuristic was implemented in C. The 32 problem instances and the example from Lee and Hsu were calculated on a Pentium PC (2 GHz). $nSucc = 6$ was selected as the maximum number of successors per node, and the time limit was set to 120 seconds.

Table II: Reference data of test cases 1 to 8 and a problem instance from Lee and Hsu (2003).

Test case	No. of stacks	Stack height	No. of container groups	No. of containers	Percentage of bay occupation (%)	Percentage of badly placed containers (%)
1	16	5	4	48	60.0	60.0
2	16	5	6	48	60.0	60.0
3	20	6	4	72	60.0	61.0
4	20	6	6	72	60.0	61.5
5	16	5	4	64	80.0	60.5
6	16	5	6	64	80.0	60.0
7	20	6	4	96	80.0	62.0
8	20	6	6	96	80.0	62.0
Lee & Hsu	16	4	4	35	54.7	37.1

The results of the test are shown in Table III. The number of moves nM of the best solution, the absolute deviation from the lower bound nM' (L_{init}), the relative deviation from the lower bound in accordance with $(nM - nM') / nM' * 100$ (%) and the required computing time in seconds are listed. For each test case the results are averaged over the appropriate four instances; finally the results are averaged over all test cases and the achieved values for the problem instance from Lee and Hsu are given. The two authors indicated a computing time of 3 minutes 15 seconds on a Pentium III processor (1 GHz).

The results in Table III 3 state that near-optimal solutions have been achieved for all the calculated instances. Optimal solutions were determined for 15 of the 32 instances, i.e. the lower bound nM' was reached. In a further 10 instances the deviation from the optimum is no more than one move, and

in the remaining 7 instances no more than two moves. Measured against the achieved test results the proposed heuristic represents an efficient procedure for the CPMP with an unspecified final layout.

Table III: Results for the test cases 1 to 8 and a problem instance from Lee and Hsu (2003)

Test case	No. of moves nM	Absolute deviation from lower bound nM'	Relative deviation from lower bound nM' (%)	Calculation time (seconds)
1	31.5	0.50	1.6	< 1.0
2	30.0	0.75	2.6	< 1.0
3	46.3	0.00	0.0	< 1.0
4	46.8	0.25	0.5	1.0
5	43.8	1.00	2.4	3.5
6	43.5	1.00	2.3	0.3
7	65.0	1.00	1.6	27.8
8	64.8	1.50	2.3	17.8
Mean	–	0.75	1.7	6.3
Lee & Hsu	15	0	0.0	< 1.0

References

- CHEN, T. (1999), *Yard operations in the container terminal – a study in the ‘unproductive moves’*, Maritime Policy and Management 26, pp.27-38
- KIM, K.H. (1997), *Evaluation of the number of rehandles in container yards*, Computers & Industrial Engineering 32, pp.701-711
- KIM, K.H.; BAE, J.W. (1998), *Re-marshaling export containers in port container terminals*, Computers & Industrial Engineering 35, pp.655–658
- KIM, K.H.; KIM, H.B. (1998), *The optimal determination of the space requirements and the number of transfer cranes for import containers*, Computers & Industrial Engineering 35, pp.427-430
- KIM, K.H.; KIM, H.B. (1999), *Segregating space allocation models for container inventories in port container terminals*, Int. J. Production Economics 59, pp.415-423.
- KIM, K.H.; PARK, Y.M.; RYU, K.R. (2000), *Deriving decision rules to locate export containers in container yards*, European J. Operational Research 124, pp.89-101
- LEE, Y; HSU, N.Y. (2002), *A network based model for the container pre-marshalling problem*, INFORMS Meeting 2002, San Jose
- LEE, Y.; HSU, N.Y. (2003), *An Optimization Model for the Container Pre-Marshalling Problem*. Preprint. Submitted to European Journal of Operational Research.
- STEENKEN, D; VOSS, S.; STAHLBOCK, R (2004), *Container terminal operation and operations research – a classification and literature overview*, OR Spectrum 26, pp.3-49.
- VIS, I.F.A; KOSTER, R. de (2003), *Transshipment of containers at a container terminal*, European J. Operational Research 147, pp.1-16

Neural Networks for Naval Applications

David E. Hess, Naval Surface Warfare Center, Maryland/USA, hessde@nswccd.navy.mil
William E. Faller, Applied Simulation Technologies, Florida/USA, faller@appliedsimulation.com
Edward S. Ammeen, Naval Surface Warfare Center, Maryland/USA, ammeeenes@nswccd.navy.mil
Thomas C. Fu, Naval Surface Warfare Center, Maryland/USA, futc@nswccd.navy.mil

Abstract

An overview is given of past and current research at the Naval Surface Warfare Center (NSWC) describing the use of neural networks (NN) for naval applications. NN Research at NSWC is conducted within the Maneuvering & Control Division, and the primary application is the development of submarine maneuvering simulation tools based on recursive neural networks (RNN). The introduction is followed by a description of the first generation RNN-based six degree-of-freedom (6-DOF) submarine simulation along with details of its performance in the ONR Maneuvering Simulation Challenge. The next section considers the case where needed input data for training is absent, and an approach is taken using feedforward neural networks as virtual sensors to intelligently estimate the missing data and thereby permit the RNN simulation to proceed. The architecture of the 6-DOF model can be modified for simulation of other vehicle types, and an example of the simulation of ship maneuvers is given. Attention has recently turned toward the use of feedforward networks to predict the time histories of force components on propulsors, and this subject is briefly addressed. Work is also underway to develop an Advanced Control and Monitoring system (ACM) which employs three state-of-the-art technologies: Robust/Reconfigurable Control, RNNs, and Fault Detection and Isolation (FDI) algorithms. This real-time system will provide vehicle monitoring, fault protection and automatic control of the vehicle from within the executive control loop. Finally, returning to the topic of submarine simulation, a second-generation RNN code is described which has been developed to support simulation-based design.

1. Introduction

The Neural Network Development Laboratory was established at the Naval Surface Warfare Center in 1994, and with the generous support of the U.S. Office of Naval Research, was directed to apply neural network technology as a predictive tool to problems of interest to the Navy. NSWC recognized that neural networks might become a useful submarine maneuvering simulation tool as a result of work using RNNs to predict and control three-dimensional unsteady separated flow fields, *Faller et al. (1995a)*, and dynamic reattachment, *Faller et al. (1995b)*. The early NSWC work concentrated on the development of an RNN-based simulation tool for 6-DOF submarine maneuvering, which has been documented in *Faller et al. (1997)*. This code performed well in the ONR Maneuvering Simulation Challenge; the results may be found in *Hess and Faller (2002)* and will be discussed in the next section. RNN simulations have been created using data from both model and full-scale submarine maneuvers. In the latter case, incomplete data measured on the full-scale vehicle was augmented by using feedforward neural networks as *virtual sensors* to intelligently estimate the missing data, *Hess et al. (1999)*. Details regarding this approach will be given below. The creation of simulations at both scales permitted the exploration of scaling differences between the two vehicles, which is described in *Faller et al. (1998)*. By adapting the submarine 6-DOF simulation and modifying the definitions of the input terms, the modeling techniques were extended to simulate ship maneuvers. An initial formulation of the problem using an RNN model for use with ships is described in *Hess et al. (1998)*. The method was then further refined and produced accurate predictions of tactical circle and horizontal overshoot maneuvers and is documented in *Hess and Faller (2000)* and will be described below. An RNN simulation with the ability to predict the 6-DOF motion of marine vehicles can then be used as a plant model within an advanced control and monitoring system. Such a system is currently under development and has been discussed in *Ammeen and Faller (2001)*, *Ammeen and Beale (2003)* and in the review paper by *Hess and Fu (2003)*. The first trial of the system, on a free-running submarine model, is expected in Fall 2004, and elements of this system will be described in a subsequent section. The topic of Real-Time Nonlinear Simulation and Design (RNS) was introduced by *Faller (2000)*. The basic idea is to use a fully trained RNN simulation operating in real-time, which takes input forces and moments acting on a marine vehicle and predicts output 6-

DOF motion. Then, alter the input forces and moments acting on the vehicle as a result of a design change to the vehicle. This input change must be described using experimental data or by using an appropriate computational fluid dynamics (CFD) code. If the latter approach is used, then one can evaluate changes to the motion of the vehicle resulting from a geometry change *without the need for any additional test data*. Early results from this work in progress are also presented below. This introduction will conclude with a brief discussion of the terminology and the neural network architecture that is common to all of the RNNs and feedforward NNs used in this work.

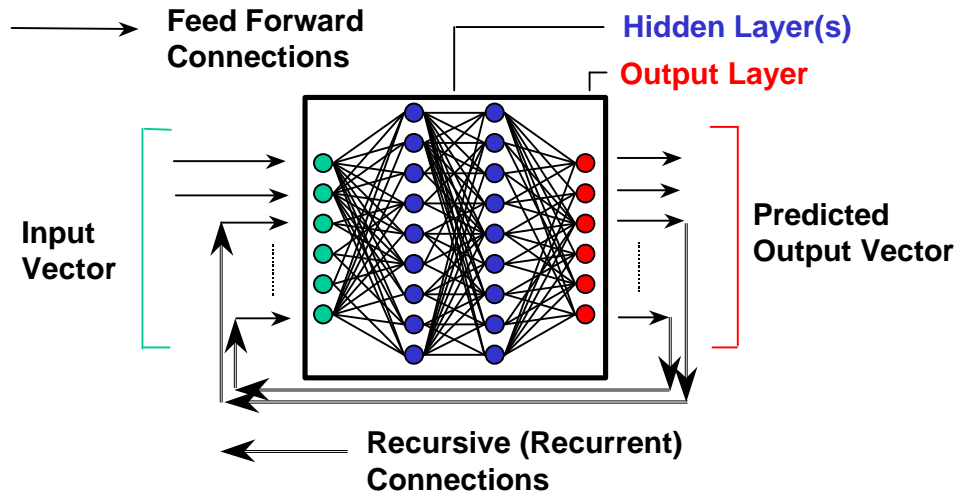


Fig.1: Neural network architecture

The basic architecture of the neural network is illustrated schematically in Fig.1. The network consists of four layers (groupings of nodes): an input layer, two *hidden* layers and an output layer. Within each layer are nodes, which contain a nonlinear transfer function that operates on the inputs to the node and produces a smoothly varying output. The binary sigmoid function is used; for input x ranging from $-\infty$ to ∞ , it produces the output y that varies from 0 to 1 and is defined by $y(x) = \frac{1}{1 + e^{-x}}$. The nodes in each layer are fully connected to those in the next layer by weighted links. As data travels along a link to a node in the next layer it is multiplied by the weight associated with that link. The weighted data on all links terminating at a given node is then summed and forms the input to the transfer function within that node. The output of the transfer function then travels along multiple links to all the nodes in the next layer, and so on. So, as shown in Fig.1, an input vector at a given time step travels from left to right through the network where it is operated on many times before it finally produces an output vector on the output side of the network.

An RNN is a computational technique for developing time-dependent nonlinear equation systems that relate input control variables to output state variables. In particular, a recursive neural network has feedback; the output vector is used as additional inputs to the network at the next time step. For the first time step, when no outputs are available, these inputs are filled with initial conditions. The time step at each iteration represents a step in dimensionless time, $\Delta t'$. Time is rendered dimensionless using the vehicle's length and its speed computed from the preceding iteration; thus, the dimensionless time step represents a fraction of the time required for the flow to travel the length of the hull. The neural network is stepped at a constant rate in dimensionless time through each maneuver. Thus, an input vector at the dimensionless time t' produces the output vector at $t' + \Delta t'$.

The process by which the network is iteratively presented with an input vector in order to produce predicted outputs that are then compared with a desired output vector is known as training. The purpose of training is to gradually modify the weights between the nodes in order to reduce the error on subsequent iterations. In other words, the neural network *learns* how to reproduce the correct answers. When the error has been minimized, training is halted, and the resultant collection of weights that have been established among the many connections in the network represent the knowledge

stored in the trained neural net. Therefore, a training algorithm is required to determine the errors between the predicted outputs and the desired target values and to act on this information to modify the weights until the error is reduced to a minimum. The most commonly used training algorithm, and the one employed here, is called backpropagation which is a gradient descent algorithm. The collection of input and corresponding target output vectors comprise a training set, and these data are required to prepare the network for further use. Data files containing time histories of control and state variables for a marine vehicle are used to effect the training.

After the neural network has been successfully trained, the weights are no longer modified and remain fixed. At this point the network may be presented with an input vector similar to the input vectors in the training set (that is, drawn from the same parameter space), and it will then produce a predicted output vector. This ability to generalize, that is, to produce reasonable outputs for inputs not encountered in training is what allows neural networks to be used as simulation tools. To test the ability of the network to generalize, a subset of the available data files must be set aside and not used for training. These validation data files then demonstrate the predictive capabilities of the network.

The authors are aware that many changes to this architecture and training process might be considered: different activation functions, different number of layers, different training algorithm, different node connections, etc. Instead, the authors have intentionally directed their efforts toward realizing applications of the technology, and therefore, this basic structure is employed throughout the work presented here. The next section will describe how this architecture is implemented into a 6-DOF submarine simulation and will give some typical examples of its performance in the *ONR Submarine Maneuvering Challenge*.

2. RNN Submarine Simulation & Results

Fig.2 shows a schematic diagram of the submarine, first developed by *Faller et al. (1997)*. The input data consist of the initial conditions of the vehicle and time histories of the control variables: propeller rotation speed and rudder and sternplane deflection angles. As the simulation proceeds, these inputs are combined with past predicted values of the state variables (outputs) to estimate the forces and moments that are acting on the vehicle. The resulting outputs are predictions of the time histories of the state variables: linear and angular velocity components which can then be used to recover the remaining hydrodynamic variables (trajectory, attitude and accelerations) required to describe the motion of the vehicle.

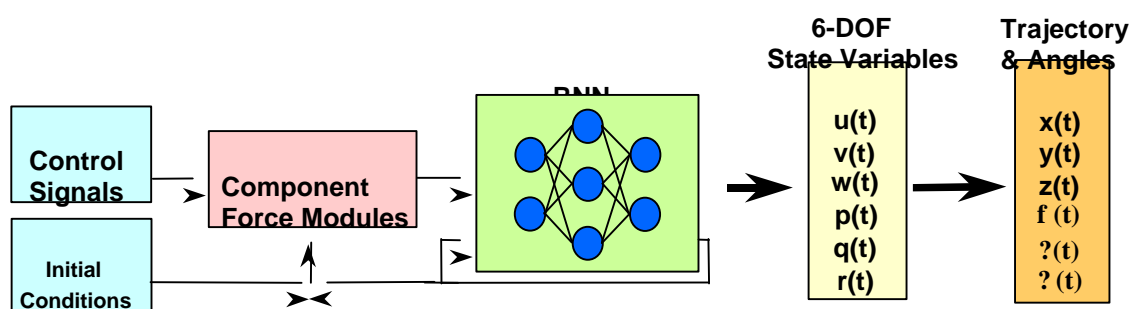


Fig.2: RNN submarine simulation

The specific terms that are modeled in the Component Force Modules box consist of seven basic force and moment terms that describe the influence of the control inputs and of time-dependent flow field effects. These include: thrust from the propeller, lift from a deflected rudder, lift from a deflected sternplane, two restoring moments resulting from disturbances in roll and pitch, and two Munk moments acting on the hull. These terms are developed from knowledge of the controls (propeller rotation speed, sternplane and rudder deflection angles), geometry of the vehicle, and from output variables, which are recursed and made available to the inputs. A detailed description of these seven inputs may be found in *Hess and Faller (2002)*.

Additional inputs are obtained by retaining past values of the seven basic inputs. This gives the network memory of the force and moment history acting on the vehicle and permits the network to

learn of any delay that can occur between the application of the force or moment and the response of the vehicle. One past value from the propeller thrust term is retained to provide an additional input. For each of the remaining 6 basic inputs, 10 past values are retained as additional inputs. The number of past values to keep is chosen empirically and appears to be a function of the frequency response of the vehicle. In this case the network is given information about past events for a period of time required for the flow about the vehicle to travel half of the vehicle length. As mentioned earlier, recursed outputs are used as six additional contributions to the input vector. Furthermore, the output vector from one previous time step is also retained and made available as six additional inputs. Knowledge of the output velocities for two successive time steps permits the network to implicitly learn about the accelerations of the vehicle.

The *ONR Submarine Maneuvering Challenge* (SMC) presented an opportunity to rigorously test the simulation and gage its response in comparison to more traditional simulation methods. The SMC engaged several participants from Government and private organizations to provide predictions of the maneuvering behavior of a radio-controlled model (RCM) submarine. The RCM conducted a variety of maneuvers in a specially equipped basin at NSWC including constant heading runs, vertical and horizontal overshoots and controlled and fixed plane turns (many with combined deflection of multiple appendages). These maneuvers, collectively, demonstrated a wide range of the maneuvering capabilities of the vehicle. A large amount of the experimental data (78 maneuvers) was provided to each participant, then each was asked to provide predictions for a separate series of 34 *blind* maneuvers. A *blind* maneuver is one for which only the initial conditions and the controls directing the vehicle are provided, and each participant was required to provide predictions of the trajectory, attitude, velocities and accelerations for the ensuing maneuver. An independent arbitrator graded each of the blind maneuvers and scores of Excellent, Good, Fair and Poor were issued.

The data for the SMC was acquired using a free-running submarine RCM funded by ONR and referred to as ONR Body 1 (OB1). The vehicle has a length of approximately 6 m with an axisymmetric hull and appendage arrangements as shown in Fig.3. The motion of the submarine is controlled by: a 3-bladed propeller, upper and lower rudders, and port and starboard sternplanes. The interior of the model is flooded with the exception of a center mounted electronics section, and the space that is occupied by the motor and control surface servo-controllers. Styrofoam and lead are installed within the hull as necessary to make the RCM neutrally buoyant with no static pitch or roll angles. All of the appendages mounted on the OB1 RCM are NACA sections, and the sail is rigidly mounted to the hull. The propeller used to drive the OB1 is a modification of a commercially available motorboat propeller. While not containing any specific attributes of actual U.S. submarines, the OB1 was designed to present computational model developers with similar types of maneuvering challenges. Hence, it remains a suitable platform for use in verifying the performance of computational tools intended to predict submarine maneuvering characteristics. Further general details on RCM operation at NSWC may be found in the report by *Nigon (1994)*.

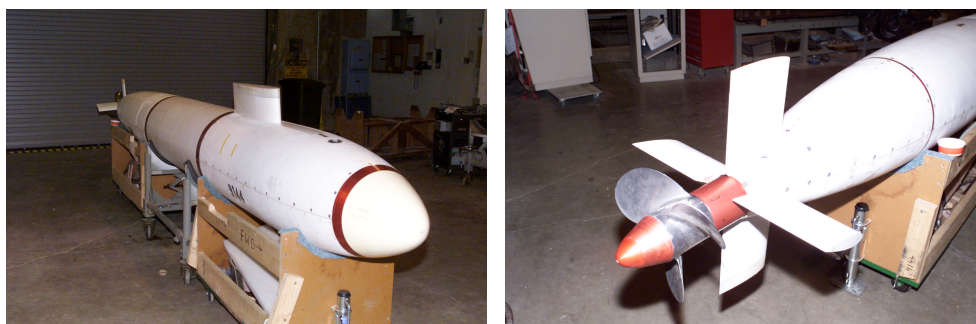


Fig.3: ONR Body 1 submarine RCM

The arbitrator graded only the *blind* maneuvers obtained from each participant, and used a grading system that relied on the Average Angle Measure (AAM). The AAM was developed by the Maneuvering Certification Action Team at NSWC in 1993-1994 in order to quantify (with a single number) the accuracy of a predicted time series when compared with an actual measured time series. The measure had to satisfy certain criteria; it had to be symmetric, linear, bounded, have low

sensitivity to noise and agree qualitatively with a visual comparison of the data. The error metric is normalized to give a value between -1 and 1 . A value of 1 corresponds to perfect magnitude and phase correlation, -1 implies perfect magnitude correlation but 180° out of phase and zero indicates no magnitude or phase correlation. This metric is not perfect; it gives a questionable response for maneuvers with flat responses, predictions with small constant offsets and small magnitude signals. Nevertheless, it is in most cases an excellent quantitative measure of agreement.

For a given maneuver, the arbitrator computed an average angle measure and a correlation coefficient for the following variables: $x-x_0$, $y-y_0$, $z-z_0$, roll, pitch, $u-u_0$, $v-v_0$, $w-w_0$, $U-U_0$, p , q , and r . The average angle results for each variable were then averaged to yield a single number for the maneuver. A grade was then assigned to that maneuver according to Table I.

Table I: Grades based on the Average Angle Measure

Excellent	> 0.9
Good	0.7 to 0.9
Fair	0.5 to 0.7
Poor	< 0.5

The RNN simulation effort received 32 Good grades and 2 grades of Fair. This was the highest total score obtained by any participant for the blind predictions. The predictions that received a grade of Fair were both constant heading runs, for which the submarine traveled in a straight path at constant speed and did not maneuver. Thus, from a maneuvering point of view, these are the least interesting. None of the participants obtained the top grade of Excellent for any maneuvers, and the overall performance by neural networks proved to be outstanding relative to that of the other contributors. Fig.4 shows overplots of predicted and measured variables for a *blind* horizontal overshoot maneuver. This particular *blind* maneuver received a grade of Good from the independent arbitrator. The only information provided to the trained RNN for this maneuver was the time histories of the controls and the initial conditions. The resulting predictions for each variable are Good to Excellent, and the average over all variables was an AAM of 0.88 and a correlation coefficient of 0.98. Thirty-two of the blind maneuver predictions were assigned a grade of Good by the arbitrator and were of similar quality. The two Fair grades occurred for constant heading runs for which the vehicle traveled in a straight line at constant speed. For these maneuvers, the RNN was required to make predictions for variables that oscillated about values very close to zero (with the exception of forward speed). As described above, the AAM metric is a poor measure for data with flat responses or small constant offsets. Examination of other error metrics for these runs demonstrated that the network performed in the Good category. In the next section we describe a procedure which was successfully used to handle a case when needed data for training was not available.

3. Neural Networks as Virtual Sensors

The preceding technique has worked well for the 6-DOF simulation of submarines at model scale for which all of the input and output variables needed to train the network are measured and are readily available. However, when conducting trials of a full-scale vehicle, the more typical case is that an incomplete set of the hydrodynamic state variables which describe the motion are measured. Specifically, the velocity components of a maneuvering full-scale submarine are difficult to measure when the vehicle is conducting submerged trials in the open ocean away from a tracking range. Despite this fundamental problem, one is motivated to try to directly simulate the full-scale vehicle because the RCM is unable to match the Reynolds numbers of maneuvers conducted with the full-scale vehicle. If one could make direct use of full-scale maneuvering data, possible scaling issues could be avoided.

The approach taken was to identify variables that *are* consistently measured on full-scale trials and use these as inputs to feedforward neural networks in order to predict the needed velocity components. To train the networks, training data was required which contained both the inputs and outputs; model

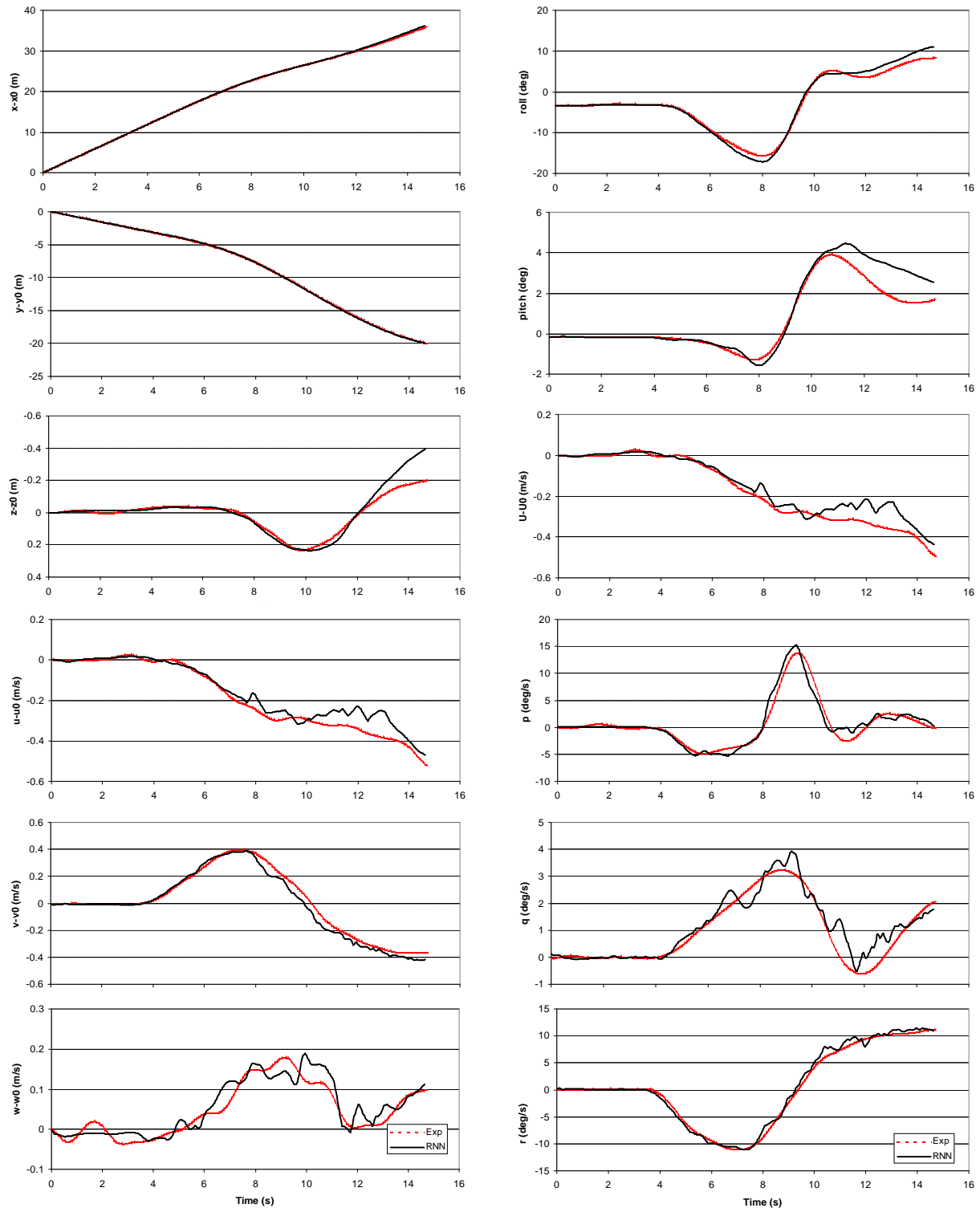


Fig.4: Blind maneuver – Horizontal overshoot

Entrance Angle = 15° , Speed = 3.0 m/s

Predictions: solid lines, Measured: dashed lines

scale data acquired from the RCM for the particular vehicle being modeled was used for this purpose. After training was completed, the neural networks contained the knowledge needed to recover the velocity components, and in that sense, they served as *virtual sensors* for the measurement of quantities, which could not be obtained by other means. Then, full-scale data, appropriately converted

to model scale, was fed into the trained networks, and predictions of the linear velocity components were obtained (and then converted back to full-scale values). Now, because the training data was at model scale, the prediction of the velocity components is fundamentally based on *the behavior of the model*. However, because the full-scale vehicle operates at higher Reynolds numbers, there is no guarantee that the transformation from inputs to outputs will be exactly the same when working with full-scale data. For that reason, a refinement procedure was employed as an additional step after obtaining the velocity predictions to assure that the velocity components were consistent with the other measured state variables at full scale. These details are described below.

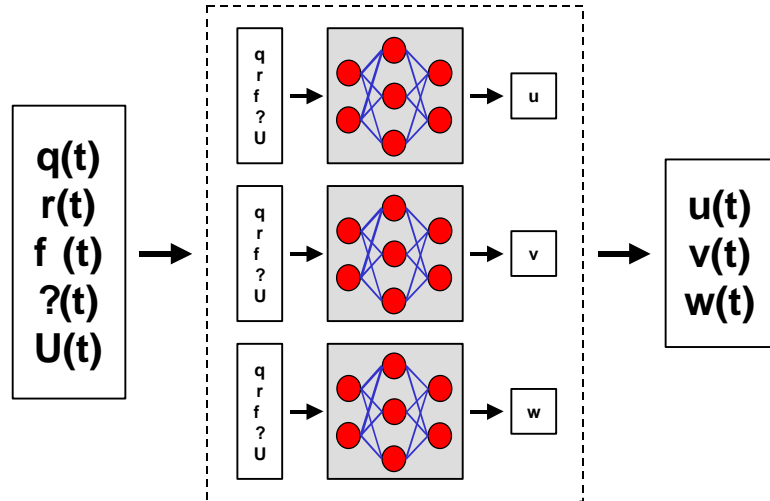


Fig.5: Schematic of neural networks for estimation of full-scale velocity components

Specifically, three neural networks were utilized – one each for the prediction of the linear velocity components u , v and w , respectively, as shown in Fig 5. Each NN used the following input quantities: pitch and yaw rates, q and r ; roll and pitch angles, j and q and speed U . The architecture of these networks consisted of a 5-node input layer, two 12-node hidden (internal) layers and a single-node output layer. All of the nodes from layer to layer were fully interconnected.

Table II: Error measures averaged over all files for the uvw-networks

Network \ Errors	Absolute Err. (ft/s)	Avg. Angle Measure	Correlation Coefficient
u	0.03	0.9996	1.0000
v	0.13	0.9799	0.9988
w	0.16	0.9546	0.9951

Error measures describing the velocity predictions prior to the refinement procedure for the 53 model-scale training and validation files are shown in Table II. In each case the average angle measure and the correlation coefficient are quite close to 1.0 indicating that the comparison of the relative shapes of the predicted and measured time histories for each velocity component is excellent. The absolute errors are reported in *ft/s* and correspond to full-scale values; these errors are extremely small.

The refinement procedure that is applied next makes use of mathematical relationships that exist between the predictions and measured full-scale quantities that *are* available for this problem. Although no tracking system is available to provide x and y , pressure sensors on the full-scale vehicle permit the measurement of depth, z . Therefore, the predicted velocity components are transformed from the body coordinate system to the inertial coordinate system using a direction cosine matrix formed from the measured Euler angles and then integrated to obtain predictions for x , y , and z . Then, the measured time history for z is substituted in place of the predicted depth and the process is reversed to obtain new predictions for u , v and w . These corrections are small with the largest correction in the w component. Next, the overall speed of the vehicle U is also a measured quantity.

Since, among the velocity components, the largest contributor to U is u , and because u is a positive-definite quantity for vehicle maneuvering, we can use U_{meas} to further improve the u component prediction by using: $u = \sqrt{U_{meas}^2 - v^2 - w^2}$. These steps ensure that the u , v and w estimates are mathematically consistent with those measured full-scale variables that exist. The minor corrections introduced by this refinement process have enhanced the u and w predictions at the expense of the v prediction; the largest remaining error is expected to be in the v component. Representative errors for the velocity components after the refinement process are given in Table III. These data are also computed from the model-scale training & validation data set because it is only for this data set that the actual values of the measured velocity components are known.

Table III: Post refinement errors

Network \ Errors	Absolute Err. (ft/s)
u	0.03
v	0.11
w	0.04

Again, the absolute errors are reported in ft/sec and correspond to full-scale values. The results indicate that the refined estimates provide improved predictions for u and w , without significantly changing the prediction of v . The post refinement linear velocity components are then used to derive the missing trajectory components and linear accelerations, and thereby complete the set of full-scale maneuvering data. This technique has not been employed when full-scale velocity data has been available, and therefore, no direct comparison between predicted velocities and full-scale velocities has been made. The point of this approach is to provide intelligent estimates of missing state variables at full scale, so that the simulation technique described in the previous section can be directly applied to full-scale maneuvering data. The fact that successful simulations have been performed at full scale, see *Faller et al. (1998)*, validates this approach. Further details concerning this technique may be found in *Hess et al. (1999)*. Returning to the RNN submarine simulation depicted in Fig.2, if the input terms describing the forces and moments acting on the vehicle are suitably altered, the simulation can be modified to predict surface ship motion. This technique is described next.

4. RNN Ship Simulation and Results

An RNN surface ship simulation was developed to predict the time histories of maneuvering variables of full-scale vehicles conducting maneuvers in the open ocean. Full-scale data describing a series of maneuvers with varying rudder deflection angles and approach speeds were acquired for each of two ships, and these data were used to train and validate several neural networks: one for each ship and type of maneuver. The input data required for the model consists of time histories of the control variables: two propeller rotation speeds and two rudder deflection angles, along with the initial conditions of the vehicle at some prescribed starting location. As the simulation proceeds, these inputs are combined with past predicted values of the outputs to estimate the forces and moments that are acting on the vehicle. As before, the outputs are predictions of the time histories of the state variables: linear and angular velocity components which can then be used to recover the remaining hydrodynamic variables required to describe the motion of the vehicle. Environmental data in the form of relative wind speed and direction are measured; future input forces and moments using these quantities are planned.

Each ship is equipped with two propellers and two rudders. One ship is larger than the other is, such that differences in size, especially with respect to the rudders and propellers, make separate simulations of interest and serve as a better test of the approach. Henceforward these ships will be referred to simply as Ship 1 and Ship 2. The neural networks have been trained to simulate two maneuvers: tactical circles and horizontal overshoots. The existing data are as follows. For Ship 1, 15 tactical circle maneuvers are available with rudder deflection angles which vary over a range of 10° to 35° and for a series of approach speeds from 5.1 m/s to 11.6 m/s (10 kn to 22.5 kn). Twenty-five

tactical circle maneuvers are available for Ship 2 with rudder deflection angles from 10° to 35° and for approach speeds from 5.1 m/s to 15.4 m/s (10 kn to 30 kn). Because ships with multiple propellers often exhibit similar turning characteristics for both right and left turns, the bulk of the data are right turns with a small number of left turns. Ten horizontal overshoot maneuvers are available for Ship 2 with rudder checking angles of 10° and 20° and for approach speeds from 3.4 m/s to 8.1 m/s (6.6 kn to 15.7 kn).

The architecture of the network is as described in Fig.1. Specifically, the input layer has 56 inputs, and each hidden layer contains 54 nodes; each of these nodes uses a bias. The output layer consists of 6 nodes, and does not use bias units. The network contains 118 computational nodes and a total of 6608 weights and biases. The bulk of the modifications to alter a submarine simulation to become a surface ship simulation occurred in the description of the forces and moments acting on the vehicle which appear in the input vector to the network. The 56 contributions that form the input vector are described as follows. Seven basic force and moment terms describe the influence of the control inputs and of time-dependent flow field effects: thrust from two propellers, T_{star} and T_{port} , lift from two deflected rudders, L_{star} and L_{port} , two restoring moments resulting from disturbances in pitch and roll, K_r and M_r , and a Munk moment acting on the hull, N_{Munk} . These terms are developed from knowledge of the controls: propeller rotation speeds and rudder deflection angles, geometry of the vehicle, and from output variables which are recursed and made available to the inputs.

As before, additional inputs are obtained by retaining past values of the seven basic inputs. One past value from each of the two propeller thrust terms is retained to provide two additional inputs. For each of the remaining 5 basic inputs, 7 past values are retained as additional inputs. The number of past values to keep is chosen empirically and appears to be a function of the frequency response of the vehicle. In this case the network is given information about past events for a period of time required for the flow about the vehicle to travel a distance of $0.63L$.

For the two surface ships modeled here, the rudders are aft of the propellers in the propeller slipstream. This is substantially different from submarines and requires modifications to the thrust and lift inputs. Therefore, the thrust input is composed of terms proportional to propeller rotation speed squared and the product of propeller rotation speed and ship speed, see *Fossen (1994)*. Then, because a deflected rudder will interact with the propeller slipstream and reduce the overall thrust imparted to the vehicle (see *Söding, 1998*), a reduction term is added based on deflection angle and thrust coefficient. The situation is similar for rudder lift. The rudder lift is composed of terms based on the local angle of attack at the rudder along with a term that accounts for enhancement of the lift from the propeller slipstream. The specific details of these terms may be found in *Hess and Faller (2000)*.

Typical trajectory predictions for tactical circles and horizontal overshoots are shown in Figs. 6 and 7 for both training and validation files. The predictions are uniformly good for the circles. For the overshoots, some drift, due to environmental effects that were not modeled, is apparent. The results may be summarized as follows. Recursive neural networks trained on tactical circle maneuvers for two separate ships were able to predict speed, trajectory components and heading with errors averaged over all the data of 5% or less. When considering only the validation maneuvers, errors for these variables ranged from 1-7%. Errors in roll estimates were from 0.2 - 0.5° for an average roll variation of 2 - 3° . For the more complex overshoot maneuver, the errors were only a little higher. Speed, longitudinal trajectory component, x , and heading exhibited errors averaged over all the data of 5% or less, whereas consideration of just the validation maneuvers increased the errors to 10% or less. Roll errors were 0.1 - 0.3° for an average roll variation of 1.2° .

The most difficult predictions for the overshoots were the transverse trajectory component, y , an integral quantity resulting primarily from the transverse velocity component, v , and to a lesser extent, the heading. The effects of wind on the overshoot maneuvers will be felt strongly in these two variables as the maneuvers were always conducted parallel or anti-parallel to the wind direction. In the case of the circles, for which environmental effects could be removed, the networks performed

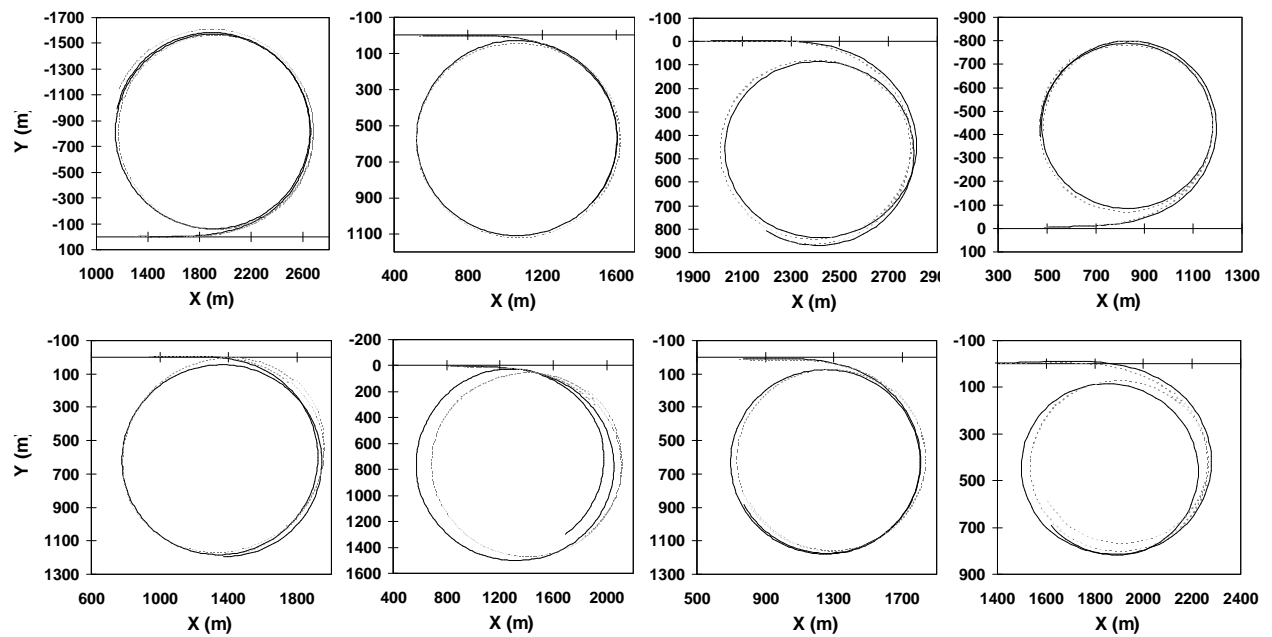


Fig.6: Ship 2 tactical circles. Top row: training files, Bottom row: validation files
Predictions: solid lines, Measured: dashed lines

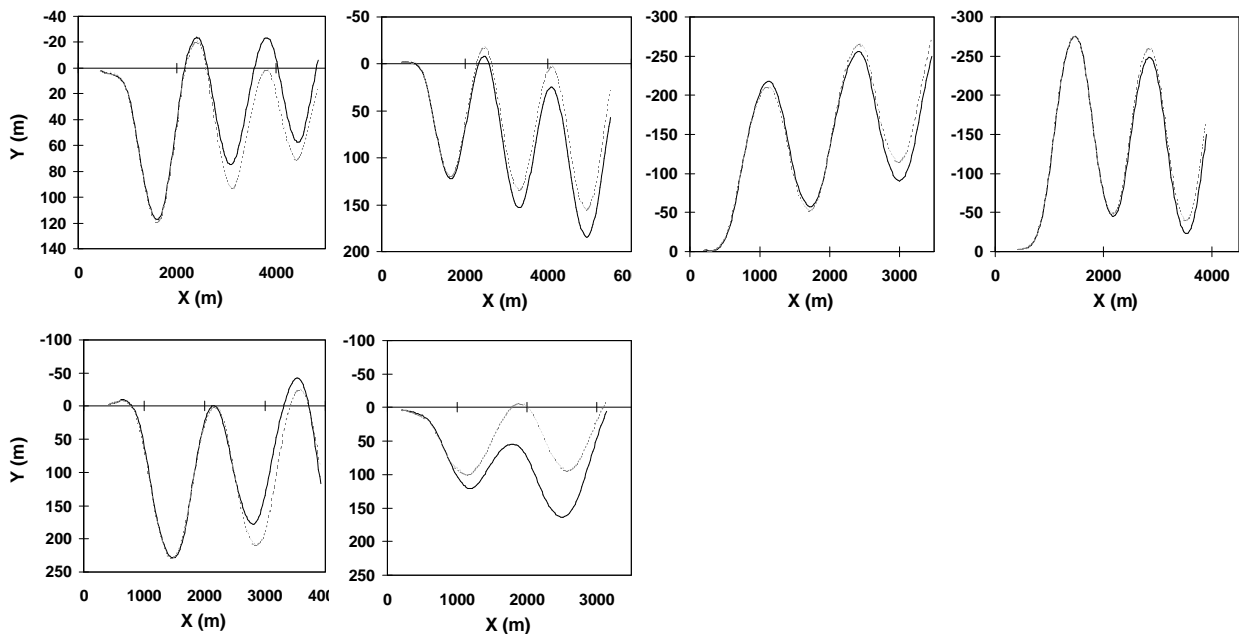


Fig.7 Ship 2 horizontal overshoots. Top row: training, Bottom row: validation files
Predictions: solid lines, Measured: dashed lines

well for these two variables. Future improvements to this simulation will include the effects of wind and wave forces and moments, and substantial improvement is likely to result. Nevertheless, at this stage of development, the simulation is predicting most of the state variables with errors of 5-7% or less. We turn now to a brief description of work in progress to use feedforward neural networks to compute force components acting on rotating propulsors.

5. Modeling Propeller Forces

Motivating the need for a propulsor force simulation is the fact that the Maneuvering & Control Dept. at NSWC is routinely required to provide *a priori* predictions of six-degree-of-freedom full-scale submarine maneuvers prior to sea trials using simulation techniques that operate in real time. Hence,

development is underway to create an improved propulsor force module that uses three feedforward neural networks to predict the time histories of three force components acting on rotating propulsors mounted on maneuvering model scale submarines. In particular, the lateral, F_y , and vertical, F_z , force components are of interest since they represent potentially large forces which can be difficult to model by other means.

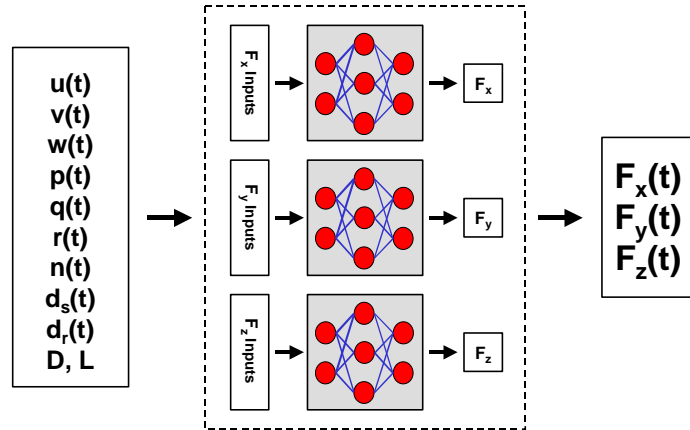


Fig.8: Propulsor force module schematic

A schematic showing the basic architecture is shown in Fig.8. A separate feedforward neural network is used for the computation of each force component. Inputs to the module are common variables readily available within simulation codes, such as the vehicle states, u, v, w, p, q, r , propulsor rotation speed, n , sternplane and rudder deflection angles, d_s, d_r , and geometrical parameters such as propeller diameter, D , and vehicle length, L . Then, internal to the module, these inputs are formed into dimensionless quantities. Inputs common to all three networks include: dimensionless speed, $U' = U/U_0$, propulsion ratio, $h = u_c/u$, propulsor inflow angle, $b = \tan^{-1}(u/0.7pnD)$, and two terms related to thrust: $J_v^{-2} = |nD/u| \cdot (nD/u)$ and $J_v^{-1} \cdot U = (nD/u) \cdot U$. Then, additional inputs are tailored to each network. For example, for the network computing the side force component, F_y , further inputs include, for example, $v' = v/U_0$, $r' = rL/U_0$, $U' \cdot v'$ and $U' \cdot r'$.

The model will first be applied to the prediction of force components on the ONR Body 1 propeller while the vehicle is conducting standard maneuvers such as horizontal overshoots, vertical overshoots, controlled turns and fixed-plane turns. Later efforts will apply the model to the prediction of propulsor forces while the model is conducting emergency recovery maneuvers such as rise jams and dive jams featuring crashback operation and deceleration of the vehicle.

The discussion to this point has concentrated on the development of advanced vehicle simulations as well as component modules to improve existing simulations. We turn attention now to an application for such a simulation. A neural network model with the ability to predict in real time the time histories of maneuvering variables of a submarine executing submerged maneuvers can then be used as a plant model within an advanced control system. A description of this work in progress is given in the next section.

6. Advanced Control & Monitoring

The goal of advanced control and monitoring in the context of marine vehicle operations is to ensure the success of the planned operation by providing the appropriate information and data to recognize and indicate anomalies in the system behavior. As a result of effective advanced control and monitoring, system reliability and operability is maximized, safety is improved, the operational envelope is expanded, and the workload of the crew is reduced. Most of the work to date has concentrated on the *monitoring* portion of advanced control and monitoring. That is, fault detection and diagnosis (classification) and the technologies to build the system. The *advanced control* portion refers primarily to *control reconfiguration* in the context of *recovery*. Although control reconfiguration is mentioned with respect to how it fits within the advanced control and monitoring

system framework, the details of its implementation are under development and will be discussed in a future paper.

The system combines three state-of-the-art techniques—Robust/Reconfigurable Control, Recursive Neural Networks (RNNs), and Fault Detection and Isolation (FDI) algorithms—into a real-time system that provides vehicle monitoring, fault protection, and automatic control of the vehicle from within the executive control loop. The combination of these technologies provides an innovative approach for the identification of both discrete and continuous failures, for differentiating between component failure and environmental influences, and for incorporating model-based fault protection into the autonomous executive control loop. The system will provide vehicle safety and operational capabilities significantly greater than can be achieved using the conventional control systems implemented today. The potential when ACM is applied to submarines is for an approximately 1000% improvement in vehicle performance during a control surface failure as measured by the vehicle change in attitude and trajectory when compared to the vehicle response obtained using a conventional control system. Such improvements can mean the difference between safe, continued operation and catastrophic failure.

The model-based components of the system, model the vehicle using recursive neural networks in order to provide a faster-than-real-time modeling and simulation capability for the vehicle reference model. The failure identification algorithms make use of the Hotelling T^2 statistic and Principal Component Analysis (PCA). PCA is employed for the reduction of the dimensionality of the data to include only those directions in the vector space that are most significant for showing variations in the training data. Fault classification has been demonstrated using both Fisher Discriminant Analysis (FDA), and Quantification of Contributing Variables (QCV). FDA requires basis (training) data for each of the fault classes. For each new data vector, a discriminant function is calculated for each of the fault classes. The class that has the largest value in the set of discriminant functions is considered to be the class that best represents that new data vector. The FDA approach is optimal in terms of separating the classes. QCV is an alternative approach to classify the detected faults. This approach neither requires nor utilizes training data for each of the fault classes, unlike FDA. Thus, fault classification can be efficiently and effectively performed without the need of an additional subsystem to generate the fault training data. The objective of this approach is to determine which observation (process) variables are most responsible for the cause of the fault, thereby assisting the fault diagnosis (classification) on the subsystems where the potential failures may occur. The fault classification is based on quantifying the contribution of each process variable to the individual scores of the principal components representing the variation in the process variables. Further details may be found in *Kim and Beale, (2001, 2002)*. The Robust/Reconfigurable control is currently based on the generic controller developed with a Linear Quadratic Regulator (LQR) solution for the gains, coupled with an on-line and real-time extension to provide simultaneous stabilization of the controller for failures and faults. Further details may be found in *Kim et al. (1997)*, *Beale and Kim (2000)*, *Ammeen (1994)*.

A simulation of a generic submarine was developed and used to test the fault detection (PCA and T^2) and classification (FDA) methodology described, *Ammeen and Beale (2003)*. The basis data from the simulation (which in this case represents the RNN) was a 12-knot combined maneuver (simultaneous 200 ft to 600 ft depth change and 120° course change). The new data (which in this case represents data from the tactical platform during operations) were generated at 12 knots and included a stern plane jam. The jammed stern plane was held constant at 15° beginning at 50 seconds. The T^2 statistic exceeded the threshold at time 51 seconds, and indicated that the stern plane jam was detected one second after the jam occurred. The basis data for the fault classes (normal operation, stern plane jam, bow plane jam, rudder jam, loss of speed sensor, and change in water density) were found to lie almost entirely in the stern plane jam confidence region. None of the data points fell within the confidence regions of the other fault classes. Since the actual fault was a stern plane jam, the fault was correctly classified. When the procedure was repeated using QCV as the classification scheme, the failure was detected and classified as a stern plane jam in $\frac{1}{4}$ of a second. Then, the procedure was repeated 100 times with the failure occurring at 100 different times along the trajectory. The results show that the failure is always detected as a stern plane jam within a maximum of $\frac{1}{2}$ second after the failure. There were no missed detections, mis-classifications, or false alarms.

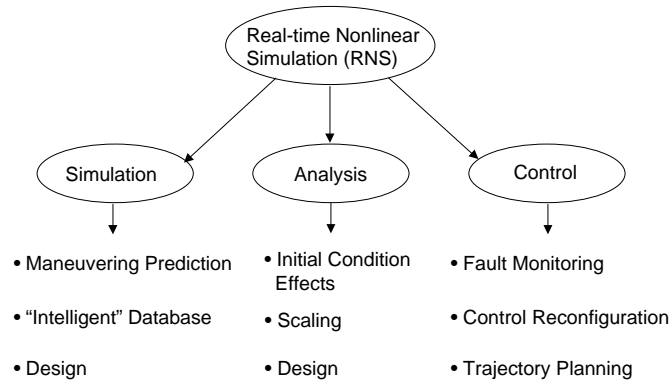


Fig. 10: Overview of RNS applications for simulation, analysis and control

developed, the RNS system can be used to explore new maneuvers months or years after the original experiment has been completed. More recently, the emphasis has shifted to the development of techniques that will facilitate the use of RNS tools in the design cycle. Such a capability would permit the entire operational envelope to be explored during the vehicle design, rather than just point conditions. For analysis, the effect of initial conditions on maneuver trajectories has been studied using an RNS system. Further, a significant effort has been made to use RNS of full-scale and model-scale vehicles to look at some of the previously unanswered questions about observed differences in maneuver trajectories. Currently, a new system that combines the strengths of a Reynolds Averaged Navier-Stokes (RANS) code and RNS into a new integrated analysis and design tool is being developed for predicting vehicle trajectories and analyzing the underlying flow fields. For control, RNS tools have been combined with existing full-scale vehicle control systems to create an integrated Advanced Control & Monitoring system, discussed in the previous section, that runs several hundred times faster than real-time. Control applications being developed include monitoring of the platform for potential failures in hardware and/or sensors and control switching schemes that can be employed to compensate for the detected failure(s). Methods for pre-determining the control sequences needed to generate a desired trajectory path are also being explored. The speed of the integrated system also permits commanded responses to be checked in order to determine, prior to execution of the maneuver, if the ordered command will exceed the available control authority of the platform and/or depart from the safe operating envelope of the vehicle.

As Fig.10 shows, RNN technologies provide a natural synthesis between laboratory experiments, experimental testing and computational fluid dynamics (CFD). A unique opportunity exists to combine several state of the art technologies to develop a new class of geometry-to-motion simulation and design tools. Currently, RNS systems can provide accurate simulations for vehicle maneuvers, including the unsteady regime. However, these systems lack geometry based inputs that would permit these approaches to be utilized in the design process. Conversely, while a number of CFD codes require only geometric inputs they tend to be best suited to predicting the steady forces on a vehicle at constant angles-of-attack and/or sideslip. Furthermore, the same CFD codes typically lack the computational speed and fidelity required to simulate the broad range of vehicle maneuvers required for design across the operational envelope. By combining the strengths of CFD and RNS a new class of integrated design tools can be developed. The CFD codes can provide both the geometric and Reynolds number information for each simulation. The RNS can provide the maneuvering simulation engine. Design tools based on this approach are envisioned to provide the following capabilities:

1. To assess the maneuvering characteristics of new designs,
2. To significantly increase the number of design alternatives which can be considered, and
3. To provide a faster than real-time geometry-to-motion simulation and design tool.

Recently, such a second-generation RNN simulation code has been developed to support submarine simulation based design. Here the focus is on the use of RNNs as a *geometry-to-motion* simulation tool. Specifically, a *parent* training data set for a particular vehicle is used to train an RNN to model how input forces and moments lead to particular output motions. The input force & moment database can originate from RCM maneuvers, captive-model experiments or RANS simulations. Design

changes to the vehicle can then be implemented by changing the input force and moment database. This can be accomplished by performing actual experiments on the modified vehicle or alternatively by performing computer simulations only. The modified force and moment database is then input into the RNN simulation to determine the design change impact on vehicle maneuvering. Since only the input force and moment database is changed, no additional training of the RNN is required. To date, this approach has been used to successfully demonstrate the maneuvering impact of design changes on the OB1 RCM.

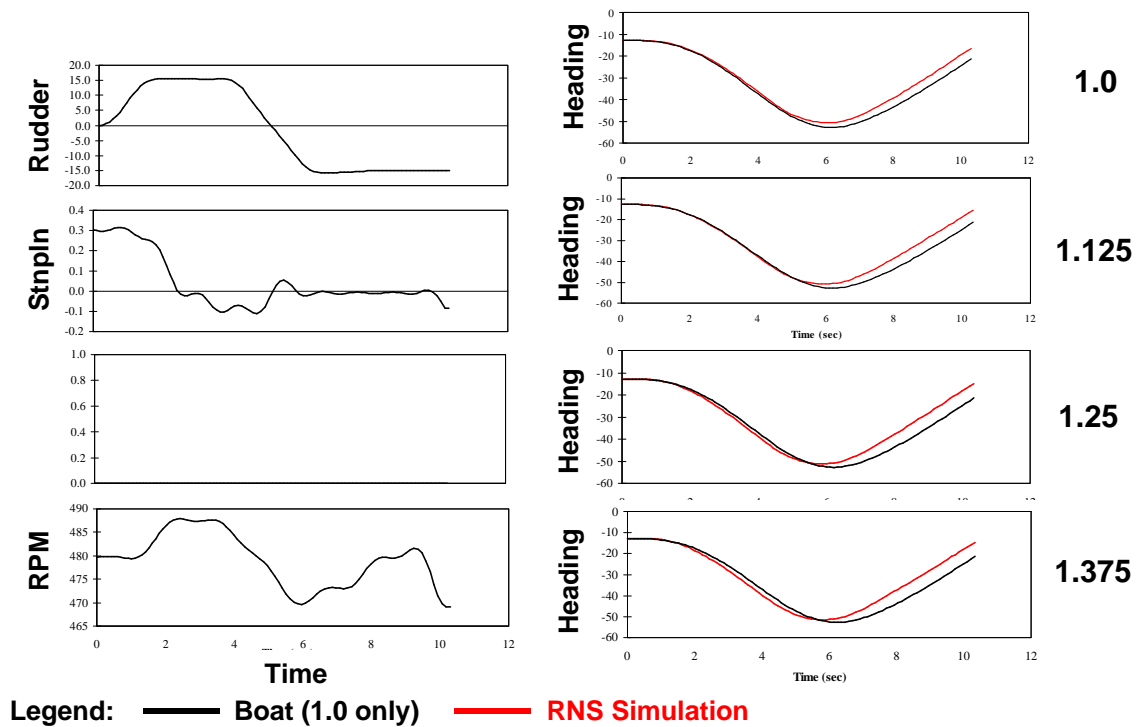


Fig.11: RNS geometry-to motion design change results

As an example, Fig.11 illustrates the maneuvering impact of changing the rudder size on OB1 as determined using this RNN design tool. The traces on the left show the control variables for this 15° horizontal overshoot maneuver. The numbers on the right are the ratio of modified rudder size to original rudder size. The lower of the two traces in the right-hand graphs is the actual vehicle heading for this maneuver (rudder size of 1.0 only) and is shown on each graph for comparison purposes. The upper traces in the right-hand graphs are the RNN prediction for each rudder size. While there is some small difference between simulation and prediction for the 1.0 case, (the simulation is not perfect), the critical information is the trend that is observed as rudder size is increased and for which no experimental data is available.

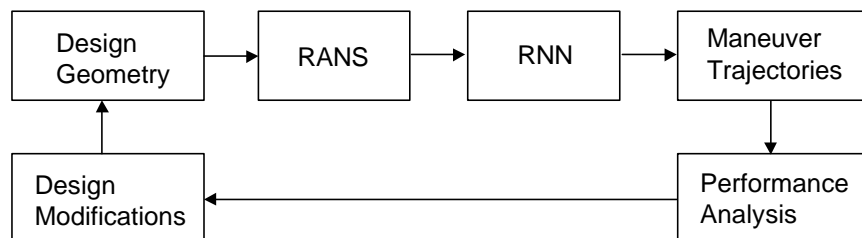


Fig.12: Integrated design process

Fig.12 shows the integrated design process under development. The information for the new submarine design is used as the input to the RANS code. The output from the RANS code is, in turn, used as the input to the RNN simulation in order to calculate the maneuvering characteristics of the new submarine design. The maneuvering behavior and hydrodynamic impacts are then analyzed,

design modifications are made, and a new geometry is supplied to the RANS code.

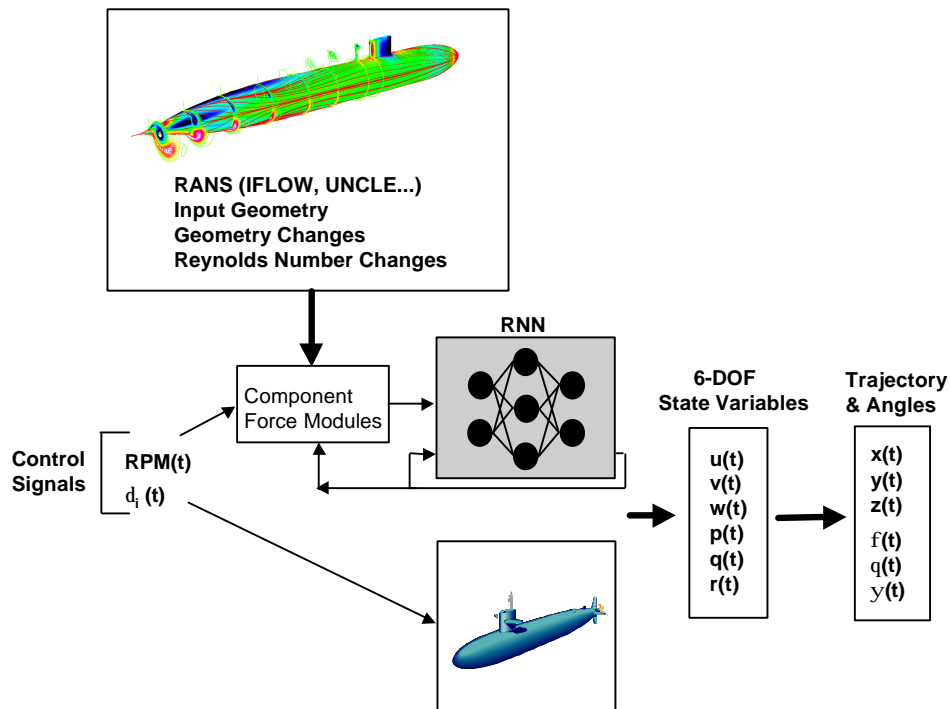


Fig.13: RNS design and simulation tool

A schematic showing an implementation of the technique using RANS input data is given in Fig.13. A series of steady RANS predictions are computed for a matrix of point conditions that bracket conditions expected during maneuvering. The RANS results are then mined to give a force and moment database that is then combined with the time-varying control signals and previous recurred outputs in order to predict the ensuing maneuver. The control inputs are the propeller rotation speed and appendage deflection angles. Since the RNN simulation engine must account for the unsteady fluid dynamics inherent to the maneuvers starting from *steady* RANS calculations, the RNS Design tool is based on a *parent* simulation. The parent simulation is developed based on *one* fully appended submarine hull configuration. This permits the unsteady hydrodynamic effects on maneuvering to be learned by the RNN and encoded within the simulation engine. After the parent simulation has been developed, no additional experimental data is required to simulate a new geometry. Any geometric changes to the hull shape, sail design and/or plane geometry is strictly dependent on performing new RANS calculations for the new design. Although not demonstrated yet, we believe that updating the RANS-calculated input terms to the design code can simulate any fully appended hull form, including radical design changes. This version of the RNS design and simulation tool using RANS input is expected to generate results in Fall 2004.

Acknowledgements

The U.S. Office of Naval Research (ONR) has generously sponsored NN and RNN development since 1994. The program officers over that time interval that have supported the work are: Dr. Patrick Purtell & Dr. Ronald Joslin in Code 333 and Dr. Theresa McMullen & Dr. Promode Bandyopadhyay in Code 342.

References

Many of these references are in electronic format and may be rapidly obtained from the authors.

AMMEEN, E.S. (1994) *Simultaneous Stabilization: A New Approach for the Design of Robust Control Algorithms*, Naval Surface Warfare Center R&D Report, CRDKNSWC-HD-1437-01, Jan.

AMMEEN, E.S.; FALLER, W.E. (2001) *Submarine advanced control and monitoring*, ASNE Intelligent Ship Symp.

- AMMEEN, E.S.; BEALE, G.O. (2003) *Advanced control and monitoring: Improving marine vehicle safety and performance*, 13th Int. Ship Control Systems Symp., Orlando,
- BEALE, G.O.; KIM, J.H. (2000) *A robust approach to reconfigurable control*, Proceedings of the Fifth IFAC Conference on Maneuvering and Control of Marine Craft, Aalborg, pp.19-202
- FALLER, W.E.; SCHRECK, S.J.; LUTTGES, M.W. (1995a), *Real-time prediction and control of three-dimensional unsteady separated flow fields using neural networks*, J. Aircraft, 32/6
- FALLER, W.E.; SCHRECK, S.J.; HELIN, H.E.; LUTTGES, M.W. (1995b), *Real-time prediction of three-dimensional dynamic reattachment using neural networks*, J. Aircraft, 32/6
- FALLER, W.E.; SMITH, W.E.; HUANG, T.T. (1997), *Applied dynamic system modeling: Six degree-of-freedom simulation of forced unsteady maneuvers using recursive neural networks*, 35th AIAA Aerospace Sciences Meeting, 97-0336, pp.1-46.
- FALLER, W.E.; HESS, D.E.; SMITH, W.E.; HUANG, T.T. (1998), *Applications of recursive neural network technologies to hydrodynamics*, 22nd Symp. Naval Hydrodynamics, Washington, D.C., pp. 708-723.
- FALLER, W.E. (2000), *Recursive neural networks: Toward advances in simulation and control*, AIAA Fluids 2000, 2000-2470, pp.1-9.
- FOSSEN, T.I. (1994), *Guidance and control of ocean vehicles*, Wiley, New York, pp. 94-96, 246-248.
- HESS, D.E.; FALLER, W.E.; SMITH, W.E.; HUANG, T.T. (1998), *Simulation of ship tactical circle maneuvers using recursive neural networks*, Workshop on Artificial Intelligence and Optimization for Marine Applications, Hamburg, pp.19-22.
- HESS, D.E.; FALLER, W.E.; SMITH, W.E.; HUANG, T.T. (1999), *Neural networks as virtual sensors*, 37th AIAA Aerospace Sciences Meeting, 1999-0259, pp.1-10.
- HESS, D.E.; FALLER, W.E. (2000) *Simulation of ship maneuvers using recursive neural networks*, 23rd Symp. Naval Hydrodynamics, Val de Reuil
- HESS, D.E.; FALLER, W.E. (2002), *Using recursive neural networks for blind predictions of submarine maneuvers*, 24th Symp. Naval Hydrodynamics, Fukuoka
- HESS, D.E.; FU, T.C. (2003), *Impact of flow control technologies on naval platforms*, 33rd AIAA Fluid Dynamics Conf., 2003-3568, Orlando, pp.1-37
- KIM, J.H.; BEALE, G.O. (2001), *Fault detection and classification in underwater vehicles using the T^2 statistic*, 9th Mediterranean Conf. Control and Automation, Dubrovnik
- KIM, J.H.; BEALE, G.O. (2002) *Fault detection and classification for underwater vehicles with quantification of contributing variables*, 10th Mediterranean Conf. Control and Automation, Lisbon
- KIM, R.R.; AMMEEN, E.S.; WARE, J.R. (1997), *Applications of neural networks and fuzzy logic in failure detection and fault tolerant control system design*, 11th Ship Control Systems Symp., Vol. 2.
- NIGON, R.T. (1994), *Radio controlled model fluid research capabilities*, Naval Surface Warfare Center R&D Report, CRDKNSWC-HD-0386-137, Jan.
- SÖDING, H. (1998), *Limits of potential flow theory in rudder flow predictions*, 22nd Symp. Naval Hydrodynamics, Washington, D.C., Vol. 2, pp.264-276

Evacuation Notation – a New Concept to Boost Passenger Evacuation Performance in the Cruise Industry

Mario Dogliani, RINA SPA (RINA), Italy, Mario.Dogliani@rina.org
Dracos Vassalos, The Evacuation Group of the Ship Stability Research Centre (SSRC),
d.vassalos@na-me-ac.uk
Tom Strang, Carnival Corp., UK, toms@carnival.com

Abstract

Breaking away from the traditional approach of the marine industry, that of passive (in-built) ship safety, RINA with assistance from the SSRC has recently developed and launched the first ever notation focused on operational issues (active ship safety); it has been tested on a ship of Carnival Cruise Lines (CCL). This Class Notation aims at assessing the effectiveness of crew functionality by comparing the evacuation performance of a ship in several specific scenarios (IMO scenarios, scenarios pertaining to social events, ship at berth and owner specific scenarios to reflect real emergencies) with and without crew assistance. This new concept makes evacuation analysis much more relevant, offering real “means” for enhancing passenger evacuation performance as well as incentivizing passenger ship owners to improve emergency procedures. This paper describes the concept and development of this groundbreaking approach, pertaining to the integration of all major crew functionalities within evacuation analysis (such as controlling spaces, searching cabins, and re-routing) and presents and discusses its implementation on a modern cruise ship, leading to conclusions and recommendations on the way forward.

1. Introduction

In the wake of the Estonia (Ro-Ro/passenger ship) disaster, trends of largely increased capacity of passenger ships, with people onboard now ranging up to 6,000, have brought the issue of effective passenger evacuation, it being the last line of defence in an emergency, to the centre of attention of the maritime industry worldwide. However, the process of evacuating a large passenger ship is a very complex one, not least because it involves the management of a large number of people on a complex moving platform, of which they normally have very little knowledge. These characteristics make ship evacuation quite different to evacuation from airplanes and buildings.

To address the risk associated with passenger evacuation at sea, the term *Evacuability* (passenger evacuation performance capability) has been devised entailing a wide range of capabilities that encompass evacuation time, identification of potential bottlenecks, assessment of layout, life saving appliances, passenger familiarisation with a ship’s environment, crew training, effective evacuation procedures/strategies, intelligent decision support systems for crisis management and design/modification for ease of evacuation. From a technical point of view, the mass evacuation of thousands of people from an extremely complex environment with unknown inaccessibility problems exacerbated by (potentially co-existing) incidents such as progressive flooding, fire/smoke and the inherent uncertainty deriving from unpredictability of human behaviour, is a problem with severe modelling difficulties at system, procedural and behavioural levels.

Evacuation has been a high priority on the International Maritime Organisation’s (IMO) agenda since 1999 when SOLAS imposed evacuation analysis to be carried out early in the design stage of new Ro-Ro passenger ships. Following this, the Fire Protection Sub-Committee, after three years of work, issued in February 2002 a set of revised Interim Guidelines for new Ro-Ro passenger ships – new cruise ships and existing ships on a voluntary basis – to be carried out either by simplified analysis or computer-based advanced analysis. Such analysis would allow for assessment at the design stage of passive safety (in-built) of the ship evacuation system only, while operational safety (active),

pertaining to any measures to enhance emergency preparedness and to better manage crisis in case of an emergency is only dealt with by means of a safety factor. In this respect, the IMO evacuation scenarios address issues relating to layout and availability of primary and secondary evacuation routes as well as passenger distribution and response times but does not address any real emergencies. Hence the need to prepare for these through better planning, training and decision support, all related to the functionality of the crew onboard, which is as crucial to passenger mustering as a good layout of the escape routes. Breaking away from the traditional approach of the marine industry to design aspects, RINA has recently developed and launched the first ever notation dedicated to operational aspects with help from SSRC and tested it on a CCL ship. This notation aims at assessing the effectiveness of evacuation procedures by comparing the evacuation performance of a ship in several specific scenarios (IMO - and ship-specific scenarios) with and without crew assistance. The final goal of this new concept is to complement the IMO evacuation analysis (which addresses mainly the design portion of ship evacuation) by specifically focusing on the efficiency of on-board procedures; used jointly with the IMO analysis, evacuation analysis would become much more relevant as well as incentivizing passenger ship owners to improve emergency procedures.

This paper describes the concept and development of this groundbreaking concept, pertaining to the integration of all major crew functionalities within evacuation analysis (such as controlling spaces, searching cabins, and re-routing) and presents and discusses its implementation on a modern cruise ship, leading to conclusions and recommendations on the way forward.

2. Evacuability

Before proceeding with the intricacies of evacuation, it is important to define the problem we try to solve and the degree to which this problem is formulated adequately for any evacuation analysis, conducted through numerical simulations, to be meaningful. In general, the ability to evacuate a ship environment within a given time and for given initial conditions (Evacuability) may be defined as follows (see Fig. 1):

$$E = f\{env, d, r(t), s(n_i); t\}$$

Thus, Evacuability is a function of a set of initial conditions, *env*, *d* and *r(t)*, and evacuation dynamics, *s(n_i)*, as explained next.

Initial Conditions: the following initial conditions (*env*, *d*, *r(t)*) should be defined and remain fixed during the execution of the simulation:

- env: ship environment model, pertaining to geometry, topology and domain semantics. For any comparisons to be meaningful we need to assume a time invariant environment for evacuation simulations. An environment changing with time (e.g., blocking doors and exits online) could not easily allow for quantifiable assessment of these effects, as it would be very difficult to repeat any such action in precisely the same state of the simulated system. However, the ability to change the environment online could offer a strong basis for crew training and for decision support in crisis management. Moreover, fire/smoke spreading and progressive flooding, the principal hazards giving rise to the need to evacuate, result in a time varying environment. Hence for any comparisons concerning global and local effects to be meaningful, any environment changes ought to be affected in a deterministic way.
- d: initial conditions of the evacuation problem, pertaining to spatial and temporal demographics of the people onboard. People in the environment will actually be randomly distributed with the possibility of fixing some initial values, e.g., placing physically disabled people on the embarkation decks and/or near an exit. As such, the initial distribution of people's demographics ought to be sampled to identify its effect on evacuability. The latter could be avoided if the distribution is known with sufficient accuracy (confidence) that a specific spatial distribution in a given time is taken to define a specific scenario for any operational or design purposes.

- $r(t)$: response time, which according to the IMO definition, is intended to reflect the total time spent in pre-evacuation movement activities beginning with the sound of the alarm. This includes issues such as cue perception provision and interpretation of instructions, individual reaction times, and performance of all other miscellaneous pre-evacuation activities. In addition, in-situ response time or any change in the state of a moving agent through intervention of e.g., crew ought to be considered. Response (awareness) time is certainly a random variable hence it has to be sampled for various distributions in order to evaluate its effect on evacuability.

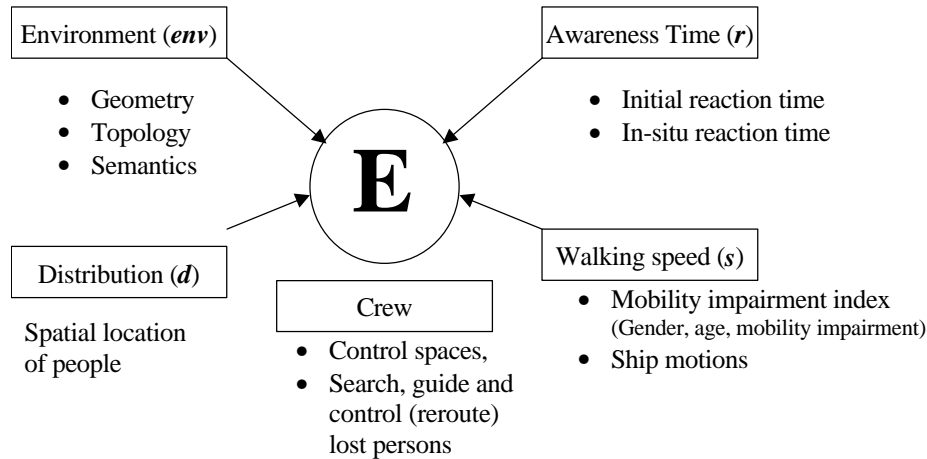


Fig. 1: The concept of *Evacuability* (E)

Evacuation Dynamics: relates specifically to walking speed, which constitutes the main motion variable of evacuation dynamics as explained next:

- $s(n_i)$: walking speed of individual flow units (agents/persons). The fact that each person onboard is dealt with as an individual flow unit and that every procedural (evacuation plan) / functional (crew assistance) / behavioural (microscopic behaviour) parameter could be accounted for as a multiplicative factor ascertaining walking speed, provides for a unique and relatively easy way for simulating evacuation, essentially being able to deal with the effect of all of these parameters by simply following a given evacuation plan, accounting for crew assistance in some agreed quantifiable way and then sample walking speed for each individual flow unit from a corresponding distribution dependent on the environment and demographics. Using the relevant mobility impairment index (MII) the walking speed in each case can straightforwardly be calculated. From a development of realistic simulation of evacuation point of view, a great deal of effort may have to be expended to accurately quantify MII for all the pertinent microscopic behaviour as well as for specific crew assistance.

On the basis of the above thinking, it may be stated that evacuability is a well-defined problem that can be formulated and solved (simulated) for given initial conditions and passenger flow parameters.

3. Crew functionality

The decision to evacuate passengers from a ship in distress is taken only as a last resort, and it is usually part of detailed contingency plans developed for modern cruise vessels as regards specific emergency situations on board including fire & explosion events, collisions, man overboard, unlawful acts, etc, *Vlaun et al. (2001)*. In cases where mustering is necessary, whilst all passengers and most of crewmembers are directed to their respective muster stations to await for further instructions, the remaining crewmembers are assigned specific emergency duties to ensure that the situation is kept under control and that passengers and crew reach assembly stations in a rapid and orderly manner; these duties include inspection of public/accommodation spaces, preparation of lifeboats/liferafts,

technical response (fire-fighting, flooding control, etc) and evacuation assistance (along the escape routes), among others. In terms of evacuation assistance (referred to also as crew functionality), the following factors are taken into account for assigning different tasks to the crew:

- the age profile of the passengers
- the degree of mobility impairment
- the varied degree of familiarity of passengers with the layout (less familiarity with secondary escape routes)
- group/friends/family behaviour
- uncertainties in human behaviour
- specific emergency procedures onboard

The last issue is of particular importance as some of the procedures may add a significant overhead to the travel time (time for reaching the assembly stations) and raise the potential for congestion. An example of this is the fact that in large cruise vessels, passengers usually must pick up their lifejackets from their cabins before proceeding to muster stations; this leads to increased counter flow and likelihood of bottlenecks on stairs. The potential negative impact associated with all other factors, are addressed through crew assistance.

It is obvious then that actual mustering on board large passenger ships may significantly deviate from the standard IMO scenarios, *IMO (2002)*, which were developed for design purposes (thus simplifying the crew role); indeed, due to differences in vessel type and operational profile, each operator has its own evacuation strategy and procedure, reflecting individual experience and operational feedback. When it comes to passenger safety, the most pro-active ship operators are now keener in having their safety policies and procedures properly quantified, recognised and acknowledged; in this respect, RINA's Class Notation is a step in this direction. In line with these developments, SSRC's evacuation simulation model *Evi* has been enhanced to explicitly accommodate all relevant crew emergency tasks (see Fig. 1) and allow quantifying their effect on the mustering process. This is done through increased interaction between the user and all the elements of the evacuation model platform, namely the passengers, crewmembers and the environment.

The new capabilities allow the assumption that a proportion of the passengers will have difficulties in finding their way to their (known) destination and are then assumed to be "lost" (it is noted that this is a conservative assumption since it is very unlikely that 30% of the passengers will be lost). Moreover, crewmembers can be assigned specific tasks, according to the procedure under analysis, like: searching passenger areas (SEARCH), directing/guiding passengers along escape routes and assisting lost passengers in finding their way to final destination, be it cabins, the closest muster station, etc. (GUIDE and CONTROL). Modelled agents (passengers or crew) can be programmed to give and receive real-time messages and to perform other specific and more complex tasks. As evacuation procedures may differ between different ship types and operators, the new capabilities are extremely useful for accurately modelling, quantifying and evaluating specific responses to any emergency scenario.

4. Measures of effectiveness of crew functionality

The evacuation performance of a ship with and without crew can be assessed on the basis of the time history of passenger Objective Completion (illustrated in Fig. 2), showing the arrival times (to their respective assembly stations) of all evacuees taking part in the simulation. In the vertical axis, the cumulative number of evacuees is shown while the time line is shown in the horizontal axis. This graph (Fig. 2) can be "normalised" with respect to the total number of evacuees, so that the cumulative percentage of evacuees that have, at a given time, reached safely at the muster station is shown in the vertical axis (Fig. 3). The *normalised* graph is referred to as the Passenger Objective Completion (POC) curve for which a number of parameters can be quantified to characterise the outcome of an evacuation simulation. Among others the following are noteworthy (see Fig. 3) as investigated by the SSRC whilst implementing crew functionality:

- The time tk : the time corresponding to the moment when a $k\%$ (k percentile) of the total number of evacuees has reached safety (the assembly station). Representative percentile values of 5, 50 and 95 can be used for comparison;
- The *area* below the POC curve, which although it has time units (average time), may be interpreted in terms of risk as the 1's complement of the normalised POC curve expresses the risk of loss of life (equivalent to the probability of an evacuee not having completed his/her objective at a given time);
- The time t_{max} : the assembly time of the last evacuee to reach safety – this time should obviously be shorter than the minimum time available before reaching untenable conditions (associated with capsizing, sinking, fire and/or smoke contamination, etc.);
- Time interval t_{95-t5} : statistic of the duration of the significant assembly process (reflecting the process duration for 90% of the evacuees) – the assembly time of the quickest and slowest 5% of the evacuees are not accounted for.

Considering that the arrival time of the evacuees is a random variable, the POC graph is different every time a new simulation run is conducted. Thus, in order to have a statistically significant result, an average (from n runs) POC curve (referred as to an APOC curve) can be obtained by calculating the average time from every percentile level of the n POC curves.

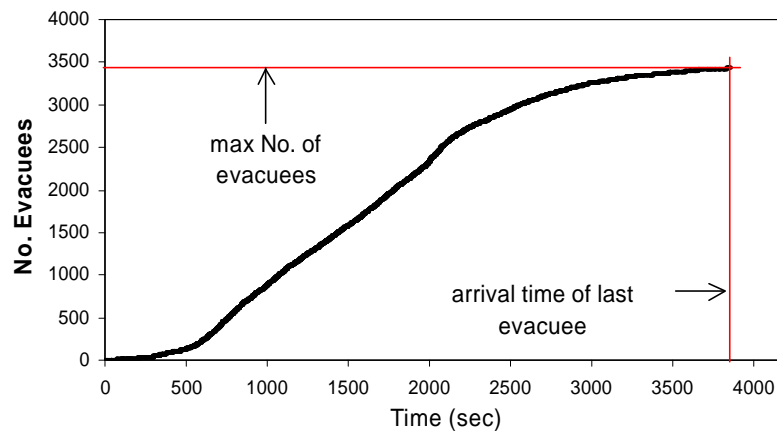


Fig. 2: Time history of passenger Objective Completion from evacuation simulation

Given that crew functionality (as implemented) relates mainly to SEARCH, GUIDE and CONTROL, and that these functions may significantly affect the shape of the POC curve – and consequently the characterising parameters, the following reasoning can be made for developing a performance criterion for evaluating the effectiveness of crew functionality:

- The POC graph and the above parameters can be calculated and compared for any evacuation scenario and for the following cases:
 - ⇒ Active crew: crew tasks, according to the evacuation procedure, are modelled
 - ⇒ Passive crew: crew are treated as normal evacuees, no interaction with passengers.
- Functions (in Evi) like SEARCH, GUIDE and CONTROL will certainly affect the *risk* of evacuees not reaching their destination on time (before a threshold time). The joint effect can be quantified by *area* and a measure of the effectiveness of crew functionality in terms of that risk can be expressed as follows:

$$\frac{t_{\max_{active}} - area_{active}}{t_{\max_{passive}} - area_{passive}}$$

- The SEARCH function determines how quickly the evacuees are alerted (minimum response time). If the minimum response time was reduced, then the assembly process would end sooner. A measure of the effectiveness of these functions at different phases of the assembly process, beginning (k=5%), middle (k=50%) and end (k=95%) of the process can be expressed as follows:

$$\frac{tk_{active}}{tk_{passive}}$$

- The CONTROL and GUIDE functions determine how fast the evacuees reach their destination (by affecting travel speed and choice of escape route) and the duration of the assembly procedure. A measure of the effectiveness of these functions can be calculated as follows:

$$\frac{t95_{active} - t5_{active}}{t95_{passive} - t5_{passive}}$$

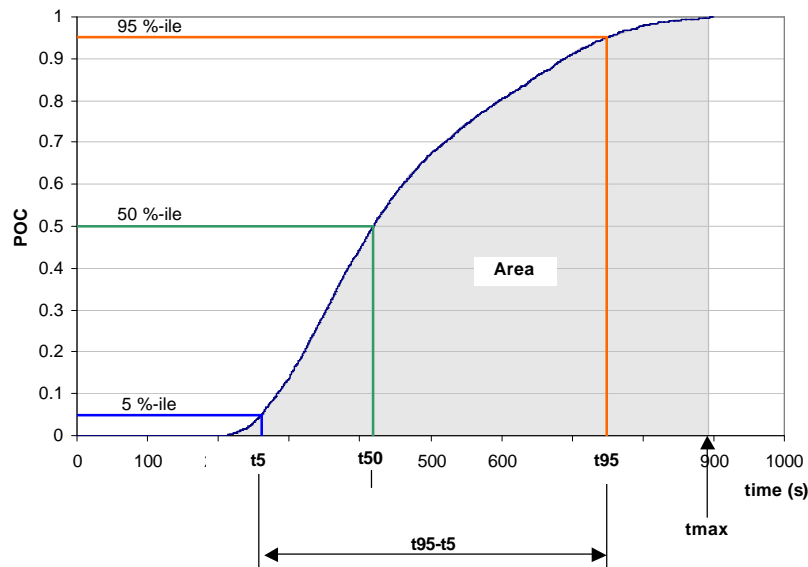


Fig. 3: Passenger Objective Completion (POC) curve – Characteristic parameters for the calculation of the crew functionality index (CFI).

In synthesis, the POC curve can be characterised by a number of parameters that expressed individually or jointly can be used to quantify – in terms of computer evacuation simulation, the assembly process in a given scenario for given initial and evolving conditions. These concepts are implicitly reflected in the performance criteria specified in the EEMS notation developed by RINA, as described next.

5. Enhanced Evacuation Management System (EEMS) – RINA notation

The basic goal of developing a class notation for evacuation was to complement the current IMO evacuation analysis requirements (MSC/Circ. 1033) and provide, once used in conjunction with IMO's analysis, a complete assessment of the evacuation safety strategy onboard consisting of a comprehensive quantitative (yet theoretical) and qualitative evaluation of the operational procedures onboard. Accordingly, the acceptance criteria (which require as a pre-requisite that the vessel fulfil IMO's analysis requirements) address the evacuation performance of the first 95% of the evacuees in a quantitative manner, and the remaining 5% of evacuees qualitatively.

5.1. Quantitative assessment

Quantitative assessment is carried out with advanced evacuation analysis (as described in MSC/Circ.1033, but enhanced to include modelling of crew functionality). The performance criteria are as follows:

- **Time Criterion (P1):** (applied to Standard Day and Night scenarios only) requires that, taking into account the active interaction between passengers and crew members, after 45 minutes (2700s) at least 95% of the passengers have reached their respective muster station¹. Thus it covers the initial and central part of the evacuation process, where most (95% of total number) of passengers (evacuees) are mustered. It can be expressed as follows (see Fig. 4):

$$POC_{active}(45') \geq 0.95 \quad \text{or} \quad t_{95} \leq 45'$$

- **Efficiency Criterion (P2):** (applied to all evaluated scenarios) requires that after 45 minutes, the area of the *normalised* POC curve (% of evacuated passengers vs. time) calculated for the *active* case is greater than the area calculated in the *passive* case. This implies that implemented crew procedures must have a positive effect after 45 minutes and can be expressed as follows (see Fig. 4):

$$E_{45'} = \frac{area_{45-active}}{area_{45-passive}} = \frac{\int_0^{t_{max}=45'} POC_{active} dt}{\int_0^{t_{max}=45'} POC_{passive} dt} > 1.0$$

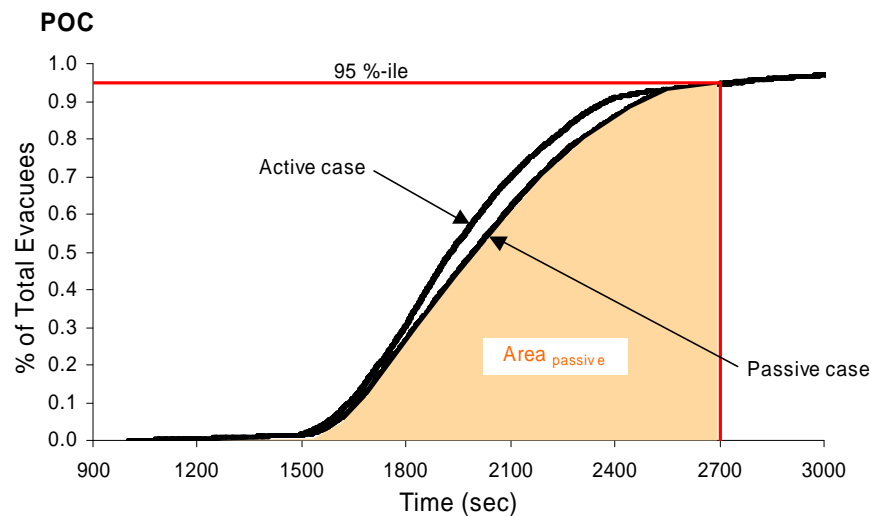


Fig. 4: Performance criterion for EEMS RINA notation

5.2. Qualitative assessment

Qualitative assessment covers the final part of the evacuation process, where it is assumed that (on the basis of operational experience) the remaining 5% of the evacuees are delayed due to unforeseen and rather specific problems (e.g. mobility impairment, panicking, etc.), which cannot be formally modelled and included in the advanced simulation analysis. Therefore the analysis is to be carried out in qualitative terms by using a checklist.

¹ Embarkation and abandonment phases are not dealt with at this stage

6. Case study

In calibrating the EEMS notation, SSRC (with the assistance of RINA and CCS) carried out more than 100 different simulations covering a wide range of situations. In the following some results are provided for the sake of illustration. These results are relevant to a large and modern cruise ship from the CCL fleet, accommodating more than 2,500 passengers and close to 1,000 crewmembers. It is noted that the aim of the analysis was to calibrate RINA's notation and was not meant to check CCL's procedures; on the contrary past experience show that CCL procedures are very effective and have therefore been used as a *best practice* on which to *tune* the EEMS notation.

Four scenarios were addressed, namely: Night standard (IMO), Day standard (IMO), Recreational Day, and Recreational Night (see Table 1). The performance criteria of the EEMS notation were evaluated by comparing two cases:

- Passive: the crew does not influence the passengers' movement, reaction time or way finding abilities, and
- Active: the crew influences the passengers' movement, reaction time and/or way finding abilities.

In both cases, 30% of the evacuees are assumed not to be familiar with the route to their destination (as mentioned earlier, this is a very conservative assumption), thus the effectiveness of crew assistance can be assessed.

Table 1: Scenarios and parameter for quantitative analysis (EEMS RINA notation)

Index	Scenario	Demographics	Awareness time	Initial distribution
1	Night – standard (IMO)	MSC/Circ 1033	MSC/Circ 1033 Uniform (600, 180)	MSC/Circ 1033 (> 2,500 pax)
2	Day – standard (IMO)	MSC/Circ 1033	MSC/Circ 1033 Uniform (300, 90)	MSC/Circ 1033 (> 2,500 pax)
3	Night – Recreational	MSC/Circ 1033	Uniform (300,90)	see description below
4	Day - Recreational	MSC/Circ 1033	Uniform (600,180)	see description below

6.1. General assumptions

The “Recreation Day” scenario was designed to simulate an at sea condition where passengers are mainly located on open decks. In this case passengers are distributed in recreational areas of open decks and in fitness areas (e.g. spa, gymnasium, beauty centre), filled to their maximum capacity as defined according to the FSS Code, and in shopping spaces and in all other areas, which are likely to be occupied during the day, e.g. libraries, children play room, etc. Crewmembers are distributed as in “Day standard” scenario.

In the “Recreation Night” scenario passengers are distributed in all recreational spaces which are likely to be occupied by night (e.g. theatre, disco, night-club, piano bar), filled to the maximum capacity determined according to the FSS Code, and among other internal and external recreational areas. Crewmembers are distributed as in Day standard scenario. This scenario was designed to assess the evacuation performance of means of escape from large internal public spaces.

Among the total crewmembers, less than 300 were explicitly modelled since they are those who, according to the procedure, carry out tasks (“Active”) significantly affecting passenger movement, reaction time and wayfinding abilities, as indicated in Table 2. The remaining crew (although contributing in real life to the safe execution of the evacuation process) are assumed not to have a direct effect on the passengers' *movement* towards muster stations and are simply modelled as defined in MSC/Circ 1033 (worth noticing, this is another conservatism in the EEMS notation).

Table 2: Summary of total and modelled **Active** crew (according to ship's Muster List)

Total active Crew members		Active crewmembers included in the simulations	
Team	Crew No.	Tasks description	Crew No.
Inspect & clear	117	Assigned to clear specific accommodation areas or spaces	83
Evacuation	215	Crew assisting passengers at muster stations (103 crew)	87
		Crew assisting passengers along escape routes at a fixed position (79 crew)	78
		Crew with special duties for evacuating critical areas or passengers, without a known position (stretcher team, youth staff) (33 crew)	-
Embarkation	176	Crew assigned to lifeboat/raft stations for preparation of the LSA	-
Technical	95	Crew in charge of dealing with the emergency (e.g. fire patrol)	-
Other	21	Crew available for performing special activities	-
Total	624	No. of crewmembers explicitly modelled	248

6.2 Discussion of results

The analysis demonstrates (even though this was not the aim of the analysis) that the evaluated procedures, applied to the vessel under analysis demonstrate a very good evacuation capability: as mentioned earlier, this was expected from past operational experience; therefore it demonstrates that the EEMS performance criteria are able to capture the essence of a good evacuation procedure.

6.2.1. Compliance with *Time Criterion* (P1)

The time criterion applies to the two Standard scenarios ("Night standard" and "Day standard"). It requires that, taking into account the active interaction between passengers and crewmembers, at least 95% of passengers be evacuated after 45' (2700s). The results obtained for the CCL ship are presented in Table 3 and Fig. 5 to Fig. 8.

Table 3: Time Criterion (P1)

Case	% of pax evacuated after 45'	CCL ship status
Night Standard	100	FULFILLED
Day Standard	96	FULFILLED

6.2.2. Compliance with *Efficiency criterion* (P2)

The efficiency criterion is to be met for all the four scenarios. It requires that after 45' the area of the normalized curve (% of evacuated passengers vs. time) calculated for the active case is greater than the one calculated in the passive case. The results obtained for the CLL ship are presented in Table 4 and Fig. 5 to Fig. 8.

Table 4: Efficiency Criterion (P2)

Case	Efficiency $E_{45'}$	CCL ship status
Night Standard	1.182	FULFILLED
Day standard	1.027	FULFILLED
Recreation day	1.027	FULFILLED
Recreation night	1.029	FULFILLED

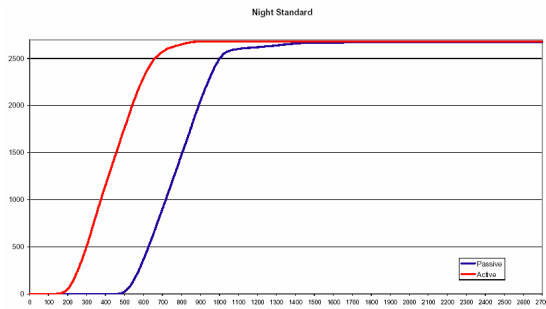


Fig. 5: Night standard case – Number of mustered passengers vs. time (45')

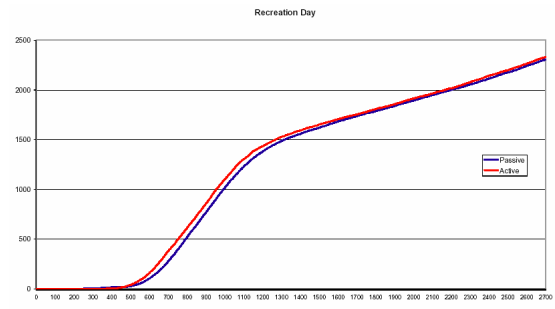


Fig. 7: Recreational Day case – Number of mustered passengers vs. time (45')

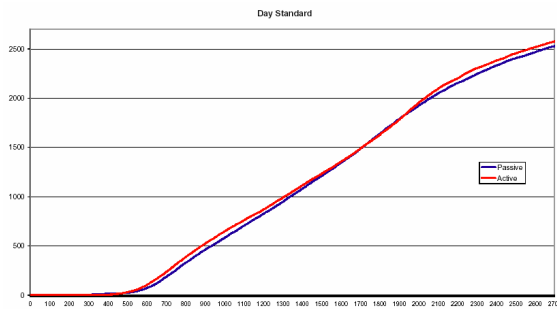


Fig. 6: Day standard case – Number of mustered passengers vs. time (45')

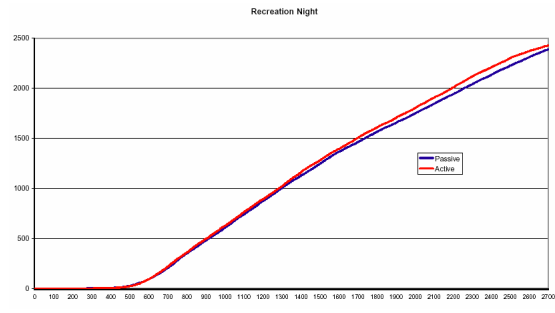


Fig. 8: Recreational Night case – Number of mustered passengers vs. time (45')

The value of efficiency obtained in the “Night standard” scenario is significantly higher than that obtained in all other cases. This is mainly due to the fact that, according to the procedure considered, the “Active” crew are alerted before the general alarm is given (5 minutes in the simulations). Note that this is a realistic, though conservative assumption; generally, when the safety officer on duty detects a potential emergency, the crisis management team, i.e. the “Active” crew is alerted and only when the potential crisis is confirmed, the general alarm is given; in other words, the “Active” crew is *activated* well before the remaining crew and passengers are alerted. Needless to mention, the 5 minutes value is a lower bound, hence the conservatism. This means that the “Active” crew is already proceeding towards (but has not necessarily reached) their assigned position on the ship before passengers and the remaining crew start moving (10' after the general alarm is activated, according to IMO criteria). However, the main reason for such a significant improvement is that, in a night case, few persons need to go back to their cabins to collect their life jackets. In all other cases the value of the efficiency is close to unity after 45'. This does not mean a limited efficiency: in several cases the parameter is well above 1.0 for long periods of time (meaning a very high efficiency of the procedure e.g. during the first half of the evacuation process) while evacuees move to the stairways and muster stations. Eventually when a large number of evacuees are closer to the muster stations the efficiency reduces due to the increased density of people entering the muster stations. This high density is however limited to the stairway enclosures in the vicinity of the muster stations, hence easily manageable by the personnel assigned in these locations (whose duty is to calm down the evacuees and, if necessary, direct them in the final few meters of the evacuation path).

In the “Recreational Night” scenario the lower value of the efficiency during the first half of the process is due, in the specific case shown in this paper, to some congestion in the proximity of the aft stairway. In real life, this congestion would be easily dealt with by re-routing some of the passengers, thus further increasing the efficiency. Nevertheless this type of action is to be decided only on a case-by-case basis by the crewmembers having the authority to decide, depending on the type of emergency. Noticeably, this is an example of the usefulness of the analyses required by the EEMS

notation: besides an objective assessment of the procedure, it leads to a useful feedback on potential critical locations onboard, thus providing a useful input for possible training of the crewmembers having the authority to decide evacuees re-routing.

Significant congestion (signified by local passenger densities being beyond 4 persons / m² for more than 10% of the assembly time, in accordance with IMO criteria) was never observed across the cases prior to the application of the full evacuation procedures.

Notwithstanding the above, when the actual procedures on board were modelled (full evacuation procedure in which passengers are prompted to return to their cabins with crew influencing the evacuation process) the main observation is the increased level of congestion due to counter flow. Indeed, in all day cases local density in many of the stairways around deck 3 were above 4 persons / m² for about 15% of the assembly time duration.

Accepting that the need for passengers to return to cabins to collect life jackets is a part of large passenger ship evacuation procedures, it is interesting to note that appropriate timing of procedures can significantly reduce levels of congestion. For the night case with crew procedures in place (Fig. 5), it is noteworthy that crew searching passenger cabins were able to clear most of the accommodation decks before “*non Active*” crew in cabins on lower decks started to fill the upper stairs. Consequently, passengers were able to travel down the stairs before being restricted by crew travelling up to their assembly station.

The size of staircases may be a contributing factor in the formation of congestion during counter flow situations. Smaller crew staircases used in evacuation may congest much more easily than larger main staircases (Fig. 9). Certainly, for very large passenger vessels, if the “return to cabins” is used, it is something that should be investigated in more detail, e.g. considering staggering the crew evacuation (i.e. pre-determined groups of “*non Active*” crew moving at slightly different times, a procedure which proved successful in High Speed Craft).

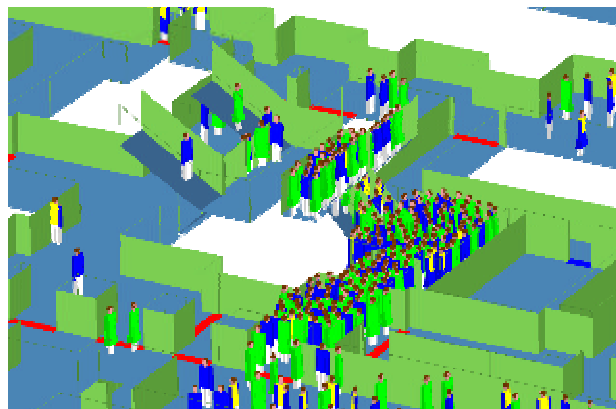


Fig. 9: Counter flow on a smaller crew staircase while the larger staircase in the adjoining fire zone remains clear.

7. Concluding remarks

Five years since the introduction of the first Interim Guidelines for a Simplified Evacuation Analysis, developments around the world concerning evacuation in the maritime context have increased in a staggering pace and research activities through multi-million projects are paving the way towards holistic approaches that address this problem as part of design/operation/training, systematically and scientifically.

When it comes to passenger safety, the most pro-active ship operators are now keener in having their safety policies and procedures properly quantified, recognised and acknowledged; in this respect, RINA's EEMS Notation is a step in this direction. In line with these developments, SSRC's evacuation simulation model *Evi* has been enhanced to explicitly accommodate all relevant crew emergency tasks and allow for quantification of their effect on the mustering process. This is done through increased interaction between the user and all the elements of the evacuation model platform, namely the passengers, crewmembers and the environment.

8. References

IMO (2002), *Interim Guidelines for Evacuation Analysis of New and Existing Passengers Ships*, MSC/Circ.1033, June 2002.

VASSALOS D., KIM H., CHRISTIANSEN G., MAJUMDER J. (2001), *A Mesoscopic Model for Passenger Evacuation in a Virtual Ship-Sea Environment and Performance-Based Evaluation*, Pedestrian and Evacuation Dynamics – April 4-6, 2001 – Duisburg.

VLAUN R., KIRKBRIDE G., PFISTER J. (2001), *Large Passenger Vessel Safety Study – Report on the Analysis of Safety Influences*, M. Rosenblatt & Son, Final Report presented to the ICCL, February 22.

RINA (2004) *CCL vessel EEMS Notation – Draft Report*

9. Acknowledgment

This study was carried out within the scope of the R&D project “SAFENVSHIP” jointly funded, under the EUREKA scheme, by a number of Companies and by several EU Member States. Their economic support is gratefully acknowledged. The authors would also like to acknowledge the significant assistance provided by personnel from both CCL and CCS (Carnival Corporate Shipbuilding): their practical and operational experience has significantly contributed to the definition of both the EEMS notation and to the tuning of the simulation software: a realistic, although notional, simulation of such a complex process as the evacuation of a large passenger ship can not be made without such type of expertise.

Neural Networks Model for Ship Maneuver

M .S. Seif, E. Jahanbakhsh, Sharif University of Technology, Seif@sharif.edu

Abstract

Maneuvering analysis is a part of design process for a ship. This issue is very important before constructing the ship as well as for its operation. Therefore, a maneuvering simulation model can be very helpful for designing and operational purposes. The ship maneuvering is a complex problem and its simulation involves many demanding fields such as free surface modeling, viscous effects, and geometrical representation of the hull and so on. Besides different shape and arrangement of ship rudder and stern part may change its maneuverability. Therefore maneuvering analysis is mostly based on experiment or approximate formulas. In the present paper, a new method is introduced for maneuvering simulation which is based on neural networks. Numerical results show good accuracy and it may be used for practical cases.

Nomenclature:

δR : Rudder angle	L: Ship length
γ : Angle between symmetry plane with X_o axes	B : Ship Breadth
u, \dot{u} : Velocity and acceleration in x direction	T : Draft
v, \dot{v} : Velocity and acceleration in y direction	Cb : Block coefficient
r, \dot{r} : Angular velocity and acceleration above z axis	m, D : Displacement
U : Ship speed	$X_o Y_o Z_o$: Inertia coordinate
V: Speed vector	xyz : Body fixed coordinate
a: Acceleration vector	MSE : Mean Square Error
C_R : Resistance coefficient	I_G : Moment of inertia

1. Introduction

The Feed Forward Neural Networks (Fen's) have been successfully applied recently to a variety of problems related with naval architecture. In fact Fen's interpolation and in some cases extrapolation capability is very powerful particularly when mapping a multi-dimensional output data space as demonstrated in Roskilly and Mesbahi (1996a). It is common for empirical data to be used directly for marine design and analysis. The non - linear functional mapping properties of Fen's and their capability to learn a new set of input patterns without significant disturbance to the previous structure are also important factors which make them particularly useful for the modeling and identification of dynamic systems as shown in Roskilly and Mesbahi (1996 b).

This brief review of the literature shows the great interest that FFN has risen recently in connection to applications ship dynamics. Moreira and Soares (2000) used a RNN model to estimate time history of ship maneuver. This model depends on time steps but the present paper explains a Neural Network

model based on FFN and RN is presented. This model does not depend on time intervals and can evaluate acceleration at any time.

The objective that of the new predictive tool is an alternative to the usual maneuvering simulators that use traditional mathematical models, which are function of the hydrodynamic forces and moments derivatives. These values are normally achieved from experiments performed which models in tanks. This procedure is time consuming and costly, requiring exclusive use of a large specialized purpose built facility. Another disadvantage of this method is the intrinsic scale effect model real ship. Anyway, this is the unique valid method that can be used in the design stage of a ship.

This paper shows the results of the comparison between simulations with a mathematical model and the simulations made with the Naval Network model after the training of data obtained through the first simulations. The purpose of this is to validate the model and to show that it is an accurate predictive tool and able to fit the results achieved with other formulation.

2. Motion Equations

The ship is assumed as a rigid body with six degrees of freedom. But maneuvering testes and simulation are usually performed in still water condition. So it is enough to consider only three degrees of freedom, is translation in horizontal plane and rotating about axis perpendicular to it. Inertia coordinate (X_0, Y_0, Z_0) and body fixed coordinate (x, y, z) are shown in Fig.1.

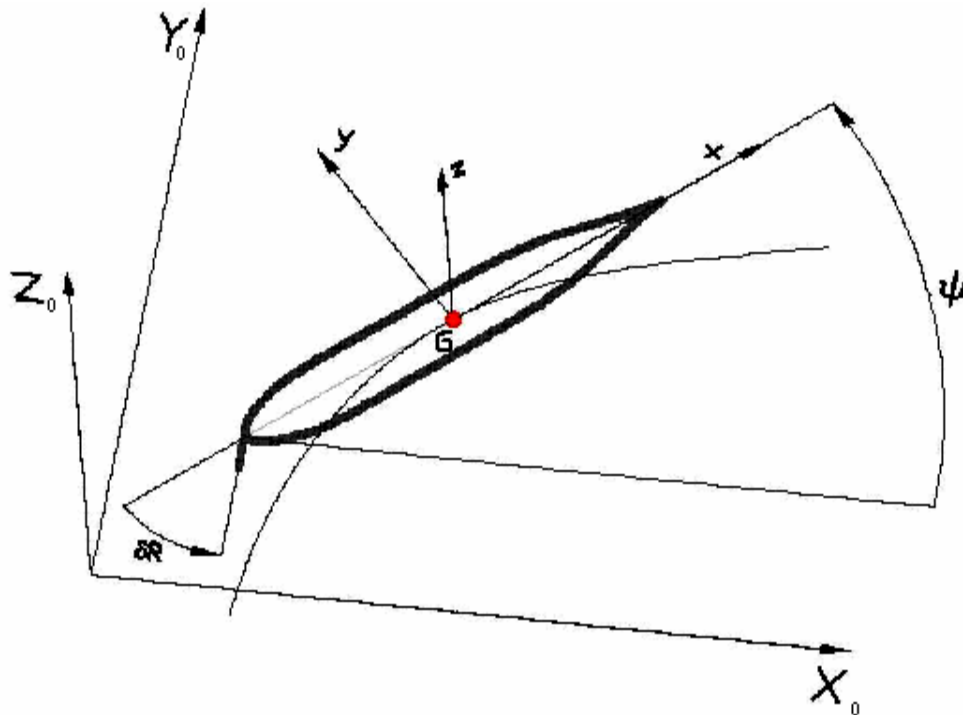


Fig.1: Coordinate Systems

Velocity vector in inertia coordinate may be written as:

$$\vec{V}_G = u\hat{i} + v\hat{j} \rightarrow \vec{V}_G = (u \cos\psi - v \sin\psi) \hat{X}_0 + (u \sin\psi + v \cos\psi) \hat{Y}_0$$

Therefore acceleration may be derived as follows:

$$\begin{aligned}
\dot{\vec{V}}_G &= \vec{a}_G = (\dot{u}\hat{i} + u\dot{\hat{i}}) + (\dot{v}\hat{j} - v\dot{\hat{j}}) = \dot{u}\hat{i} + u\dot{\vec{y}}\hat{j} + \dot{v}\hat{j} - v\dot{\vec{y}}\hat{i} \\
&= (-v\dot{\vec{y}} + \dot{u})\hat{i} + (u\dot{\vec{y}} + \dot{v})\hat{j} \\
\begin{cases} a_x = \dot{u} - v\dot{\vec{y}} \\ a_y = \dot{v} + u\dot{\vec{y}} \end{cases} &\xrightarrow{r=Y} \begin{cases} a_x = \dot{u} - vr \\ a_y = \dot{v} + ur \\ \mathbf{a} = \dot{\mathbf{r}} \end{cases} \quad 1
\end{aligned}$$

Newton's second law yields motion equation for the ship in three degrees of freedom as:

$$\begin{aligned}
\sum F_x &= ma_x \\
\sum F_y &= ma_y \\
\sum I_z &= I_G \mathbf{a}
\end{aligned}$$

It is usual to replace $\sum F_x$ by X , $\sum F_y$ by Y and $\sum M_z$ by N . These forces are mostly hydrodynamic forces and different mathematical or experimental methods are used for their evaluation. A linear model is used in this paper as follows:

$$\begin{aligned}
X &= X(u, v, r, \dot{u}, \dot{v}, \dot{r}, dR) \\
Y &= Y(u, v, r, \dot{u}, \dot{v}, \dot{r}, dR) \\
N &= N(u, v, r, \dot{u}, \dot{v}, \dot{r}, dR)
\end{aligned} \quad 2$$

Using Taylor expression for the above function with the assumptions:

$$u = U, \quad \dot{v} = 0, \quad \dot{r} = 0, \quad \dot{u} = 0, \quad \dot{v} = 0, \quad \dot{r} = 0, \quad dR = 0$$

Yields:

$$\begin{aligned}
X &= \frac{\partial X}{\partial u}(u - U) + \frac{\partial X}{\partial v}v + \frac{\partial X}{\partial r}r + \frac{\partial X}{\partial \dot{u}}\dot{u} + \frac{\partial X}{\partial \dot{v}}\dot{v} + \frac{\partial X}{\partial \dot{r}}\dot{r} + \frac{\partial X}{\partial dR}dR + X(u = U) \\
Y &= \frac{\partial Y}{\partial u}(u - U) + \frac{\partial Y}{\partial v}v + \frac{\partial Y}{\partial r}r + \frac{\partial Y}{\partial \dot{u}}\dot{u} + \frac{\partial Y}{\partial \dot{v}}\dot{v} + \frac{\partial Y}{\partial \dot{r}}\dot{r} + \frac{\partial Y}{\partial dR}dR + Y(u = U) \\
N &= \frac{\partial N}{\partial u}(u - U) + \frac{\partial N}{\partial v}v + \frac{\partial N}{\partial r}r + \frac{\partial N}{\partial \dot{u}}\dot{u} + \frac{\partial N}{\partial \dot{v}}\dot{v} + \frac{\partial N}{\partial \dot{r}}\dot{r} + \frac{\partial N}{\partial dR}dR + N(u = U)
\end{aligned} \quad 3$$

It is customary to use following notation to simplify the Eq.3.

$$X_u = \frac{\partial X}{\partial u}, \quad X_v = \frac{\partial X}{\partial v}, \quad X_r = \frac{\partial X}{\partial r}, \quad \dots \quad 4$$

Ship hull is symmetric about xz plane and some terms like,

$$N_{\dot{u}}, \quad N_u, \quad Y_{\dot{u}}, \quad Y_u, \quad X_{dr}, \quad X_{\dot{r}}, \quad X_r, \quad X_{\dot{v}}, \quad X_v, \dots \text{ may be neglected [14].}$$

By defining nondimensional coefficients, motion equations become:

$$\begin{aligned}
X' &= X'_u(u - U) \\
Y' &= Y'_uv + Y'_rr + Y'_v\dot{v} + Y'_r\dot{r}
\end{aligned} \quad 5$$

$$N' = N'_u v + N'_r r + N'_\dot{v} \dot{v} + N'_\dot{r} \dot{r}$$

Nondimensional coefficients are as follows:

$$X' = \frac{X}{\frac{1}{2} \rho U^2 L T}$$

$$Y' = \frac{Y}{\frac{1}{2} \rho U^2 L T}$$

6

$$N' = \frac{N}{\frac{1}{2} \rho U^2 L^2 T}$$

Mathematical methods and experiments are used for evaluating hydrodynamic coefficients. But, for preliminary design stages, some approximate formulas are popular. These formulas are based on available results and data. Typical formulas are resented in Eq.7 [14].

$$Y'_r = -\rho \left(\frac{T}{L} \right)^2 [0.67 B / L - 0.0033 (B / T)^2]$$

$$N'_u = -\rho \left(\frac{T}{L} \right)^2 (1.1 B / L - 0.041 B / T)$$

7

$$Y'_v = -\rho \left(\frac{T}{L} \right)^2 (1 + 0.4 C_B B / T)$$

$$Y'_r = -\rho \left(\frac{T}{L} \right)^2 (-1/2 + 2.2 B / L - 0.080 B / T)$$

$$N'_v = -\rho \left(\frac{T}{L} \right)^2 (1/2 + 2.4 T / L)$$

$$N'_r = -\rho \left(\frac{T}{L} \right)^2 (1/4 + 0.039 B / T - 0.56 B / L)$$

For X'_u resistance coefficient C_R is used Motion equations are derived by substituting Eq.7 in Eq.4. General procedure for evaluating ship path is represented in Fig.2.

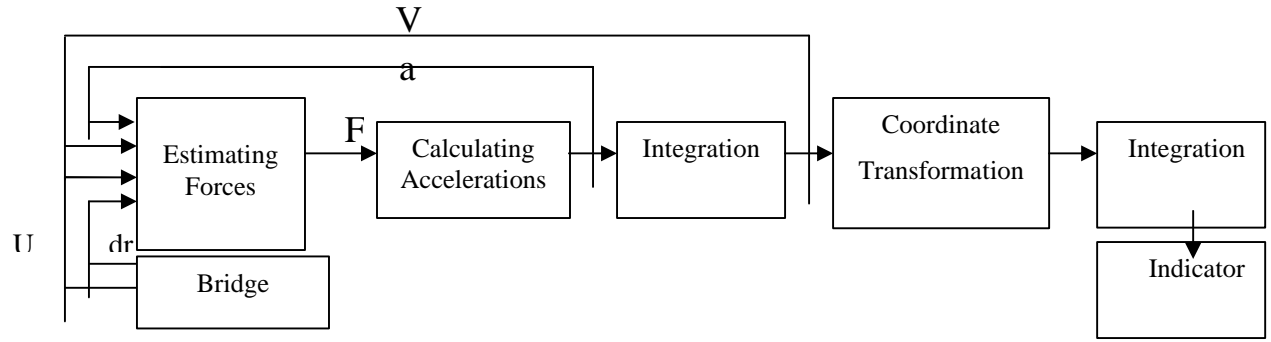


Fig. 2: Mathematical approach for simulating ship maneuver

Forces are calculated based on speed and acceleration which are feedback signals and bridge command. (Fig.2). Using motion and calculated forces responses are evaluated. Since Speeds and accelerations are found in body fixed coordinate, a transformation is required to the inertia coordinate. Ship path is also defined in this coordinate.

3. Neural Networks Model

The first application of neural networks returns back to 1950. But it has found many applications in recent years and these include signal processes, time series forecasting, modeling and control fields [15].

In the present paper a new model is introduced for ship maneuvering simulation. It uses Error Back Propagation (B-P). The inputs are velocities, rudder angle and sailing speed. Outputs are accelerations and can be used to find new position of the ship.

The motion equations are considered as follows:

$$X = m(\dot{u} - vr)$$

$$Y = m(\dot{v} + ur) \quad 8$$

$$N = I_z \dot{r}$$

Combining (2) and (8) yields:

$$X = X(u, v, r, \dot{u}, \dot{v}, \dot{r}, \mathbf{dR}, U) = m(\dot{u} - vr) \quad 9$$

$$Y = Y(u, v, r, \dot{u}, \dot{v}, \dot{r}, \mathbf{dR}, U) = m(\dot{v} + ur)$$

$$N = N(u, v, r, \dot{u}, \dot{v}, \dot{r}, \mathbf{dR}, U) = I_z \dot{r}$$

The above equations may be simplified to the following from:

$$\dot{u} = f_1(u, v, r, \dot{v}, \dot{r}, \mathbf{dR}, U) \quad 11$$

$$\dot{v} = f_2(u, v, r, \dot{u}, \dot{r}, \mathbf{dR}, U) \quad 12$$

$$\dot{r} = f_3(u, v, r, \dot{u}, \dot{v}, \mathbf{dR}, U) \quad 13$$

It is possible to introduce new relations based on (10), (11) and (12):

$$\dot{u} = h_1(u, v, r, \dot{r}, \mathbf{dR}, U) \quad 14$$

$$\dot{r} = h_2(u, v, r, \dot{u}, \mathbf{dR}, U) \quad 15$$

Or finally \dot{u} can be presented in the following form:

$$\dot{u} = g_1(u, v, r, dR, U) \quad 15$$

For v and r similar functions may be introduced as follows:

$$\dot{v} = g_2(u, v, r, dR, U) \quad 16$$

$$\dot{r} = g_3(u, v, r, dR, U) \quad 17$$

These equations are relationships among velocities and accelerations. There fore, it is possible to use neural networks model to represent the behavior of these variables. A BP networks is employed. The inputs are u, v, r, dR , u and outputs are \dot{u}, \dot{v} and \dot{r} . The simple modeling procedure is shown in Fig.3.

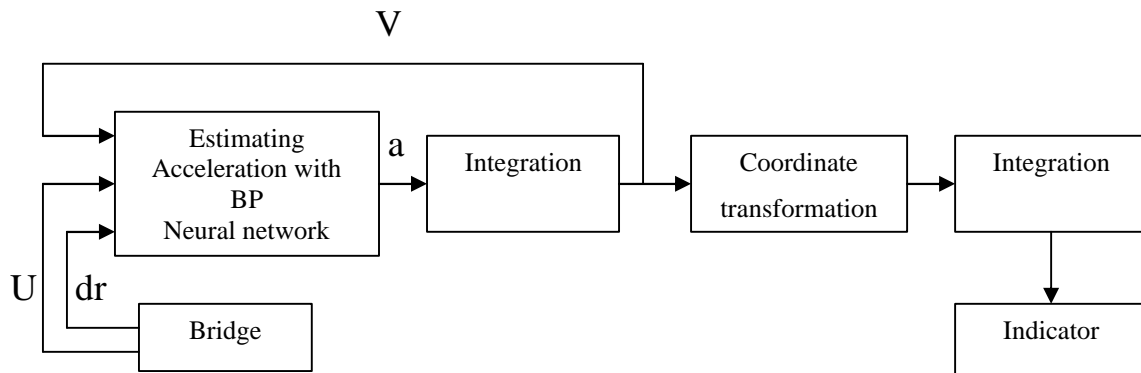


Fig.3: Neural Networks approach for simulating ship maneuver

No.	Input Patterns					Output Patterns		
	u	v	r	dr	U	\dot{u}	\dot{v}	\dot{r}
1								
3000								

Fig. 4: Format of patterns used for training

The mathematical model which explained in previous section is used for training of networks. 3000 patterns are used in training procedure. These are in the general form shown in Fig.4. Training procedure is based on Levenberg – Marquardt [16]. In Fig.5, variation of MSE for the networks is presented.

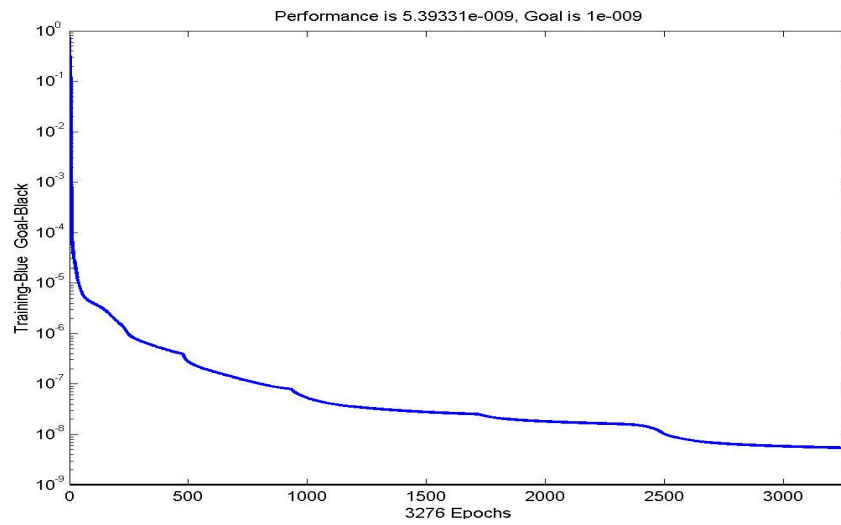


Fig. 5: Error variation for the networks

4. Numerical Results

The model is used for simulating of a ship in three different maneuvers. Principal particulars of the ship are given in the following table.

L	152.05 m
U	14.62 knot
B	20 m
T	9.639 m
Cb	0.64

The model is trained by patterns form mathematical model. Then mathematical model and networks are applied for new conditions and results are compared.

1) Turning Maneuver

The case can show the area or diameter necessary for ship turn. Fig. 6 shows results for a typical case. Rudder angle is considered to be 17.5 degrees and the speed of the ship is 14.62 knots.

It is clear That both methods give very close results and the maximum difference found to be about 1 m after 360° turn.

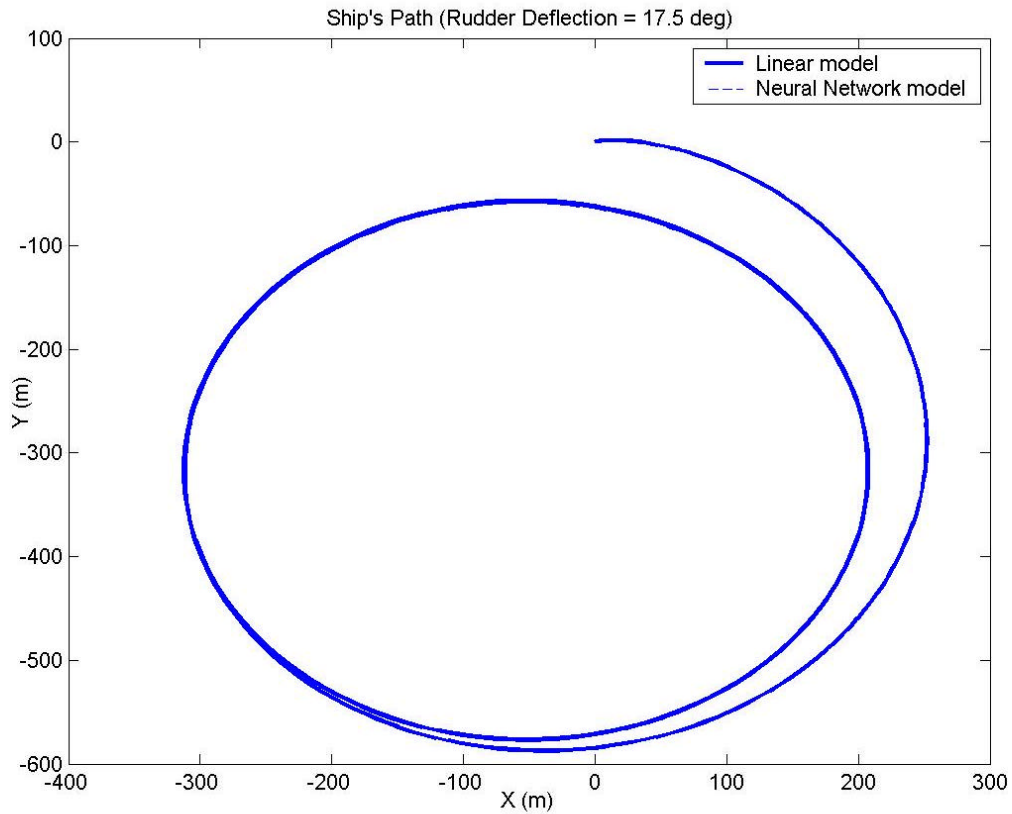


Fig 6: Turning maneuver

2) Zig Zag test

The control performance of a ship is very important. It is mainly depends on ship rudder and can be evaluated through ZigZag test. In this case study, rudder is assumed to have behavior given in Fig.7. The angle is changed from 7.5 degrees starboard to 7.5 degrees port regularly and in each side it will be stopped for 50 seconds. This is a typical ZigZag test procedure for the ship.

Fig. 8 shows Ship's trajectory. Both methods give very similar results. The difference in the position of ship will be about 50 m after 12 km and it can be considered satisfactory for practical applications.

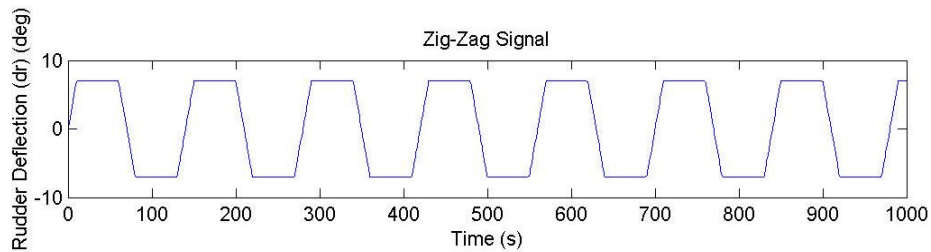


Fig. 7: Rudder angle in Zig Zag test

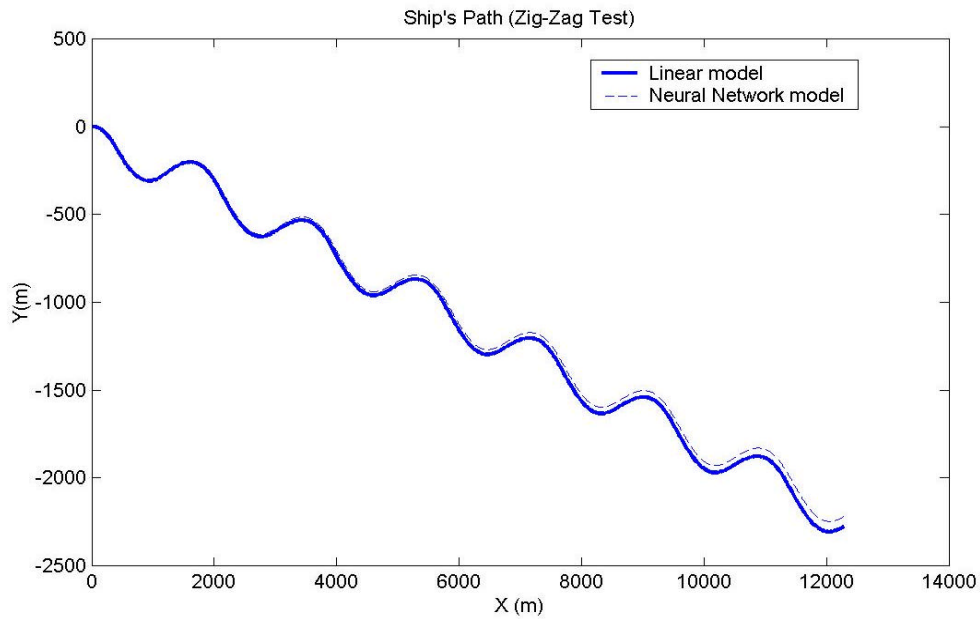


Fig 8: Ship's path in Zigzag test

3) Irregular Maneuver

In a practical case, the ship may enter an irregular maneuver. It means according to the conditions, rudder may be positioned in different condition for different time intervals. In this case which can show a complex behavior, Fig.9 is assumed for rudder angle change. The simulation results are also presented in Fig.10. Even in this irregular maneuver, results show very similar trend. Such results are very valuable for simulation of ship maneuver in ports or channels

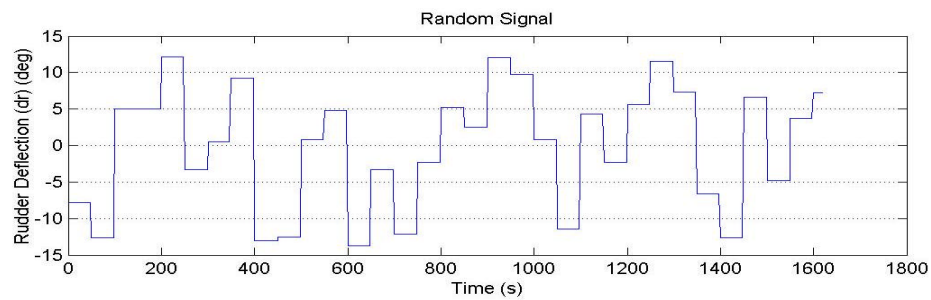


Fig. 9: Rudder angle in irregular maneuver

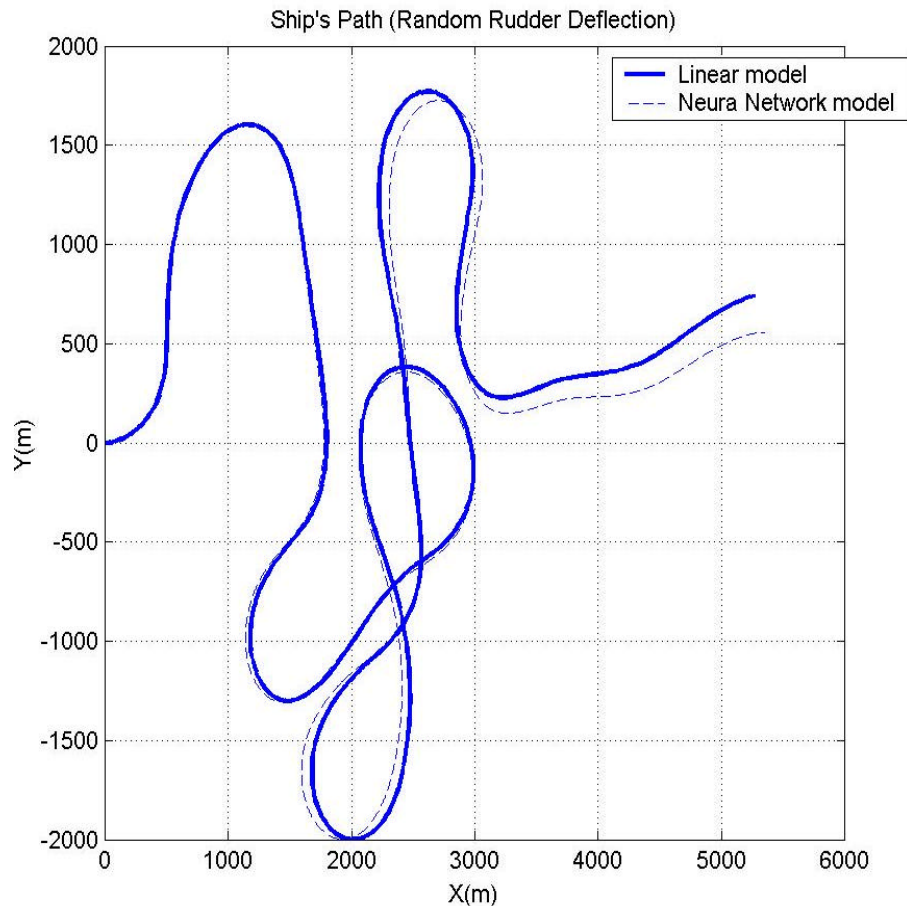


Fig. 10: Ship's path in irregular maneuver

5. Conclusions

Mathematical models for ship maneuvering simulation are based on hydrodynamic coefficients. Such coefficients are evaluated mostly in experimental or approximate forms. Besides some simplification in motion equations are made and the exact motions may not be derived. Therefore a neural networks model may be used as a reliable tool for such cases. If the experimental results or data from full scale trials are sufficient, it is possible to train the networks properly and use it for different purposes. The BP model is used in this paper and it proved a good accuracy. The possibility for adding other external loads such as wind, currents and so on is also taken into account.

References

- [1] NOMOTO, K., TAGUCHI, K., HONDA, K. and HIRANO, S., "On the Steering Quality of Ships," International Shipbuilding Progress, Vol. 4, pp. 354-370 (1957).
- [2] DAVIDSON, K. S. M., SCHIFF, L. "Turning and Course-Keeping Qualities", Transactions SNAME, 1946.
- [3] NORRBIN, N.H., "Theory and Observation on the Use of a Mathematical Model for Ship Maneuvering in Deep and Confined Waters," Swedish State Shipbuilding Experimental Towing Tank, Publication 68, 1971.

- [4] ALMAN, P.R., BERTSCHE, W.R., BOYISTON, J.W., CARD, J.C., CRANE JR., J.L., EDA, H., KEITH, V.F., LANDSBURG, A.C., MCCALLUM, I.R., MILLER, JR., E.R., and TAPLIN, A., "Design and Verification for Adequate Ship Maneuverability, etc." *SNAME Transactions*, Vol.91.
- [5] OLTMANN, P., AND SHARMA, S.D., "Simulation of Combined Engine and Rudder Manueveres Using an Improved Model of HullPropeller-Rudder Interactions," *Proceedings*, 15th ONR Symposium on Naval Hydrodynamics, Hamburg, 1984.
- [6] CLARKE, D., "Assessment of Maneuvering Performance," *Ship Maneuverability Prediction and Achievement*, RINA, London, 1989.
- [7] INOUE, S., HINARO, M., KIJIMA, K. AND TAKASHINA, "A Practical Calculation Method of Ship Maneuvering Motion ." *International Shipbuilding Progress*, Vol. 28, No. 321, 1981.
- [8] POURZANJANI, M., ZIENKIEWICS, H., AND FLOWER, J. O., "A Hybrid Method of Estimating Hydrodynamically Generated Forces for Use in Maneuvering Simulation." *International Shipbuilding Progress*, Vol.34, No. 399, 1987.
- [9] CRANE, C.L., EDA, H., LANDSBURG, A., 1989. Controllability. In: Lewis, E.V. (Ed.), *Principles of Naval Architecture*, vol. 3. SNAME, Jersey City, pp. 191–365.
- [10] INOUE, S., HINARO, M., KIJIMA, K. AND TAKASHINA, "Hydrodynamic Derivatives on Ship Maneuvering." *International Shipbuilding Progress*, Vol. 28, No. 321, 1981.
- [11] ATLAR, M., MESBAHI, E., ROSKILLY, A.P., GALE, M., 1998. Efficient techniques in time-domain motion simulation based on artificial neural networks. In: *International Symposium on Ship Motions and Maneuverability*, RINA, London, UK,, pp. 1–23.
- [12] HADDARA, M.R., WANG, Y., 1999. On parametric identification of maneuvering models for ships. *Int. Shipbuilding Prog.* 46 (445), 5–27.
- [13] MOREIRA, L., GUEDES SOARES, C., 2002. Mathematical models for ship path prediction in maneuvering simulation systems. *Ocean Eng.* 29 (1), 1–19.
- [14] LEWIS, E. V., *Principle of Naval Architecture*, Second Revision, Vol. III, The Society of Naval Architectures and Marine Engineering, 1989.
- [15] HAYKIN, S., *Neural Networks, A Comprehensive Foundation*, New York MacMillan Publishing Company, 1994.
- [16] MATLAB Manual

Correlation Strategy for Propeller Excited Pressure Fluctuations

Wilfried Abels, Stefan Krüger, TU Hamburg-Harburg, w.abels@tu-harburg.de

Abstract

Market demands have lead to a significant improvement of the vibration and comfort level, especially for complex ship types such as RoRo and RoPax ships. The need to optimise the design for efficiency does not allow to spend extra material with the single purpose to keep the vibration levels at the required values. The better strategy is to minimize the exciting forces as such, where the propeller is the most important source for discomfort, noise and vibration. Whenever a new ship design had vibration problems, the propeller could be identified as the main reason. Changes related to the propeller are a substantial cost factor. To minimize the propeller impact on the hull, the wake field has to be optimised on one hand and the design of the wake adapted final propeller on the other. The demand for low pressure pulses has lead to propeller designs characterized by low cavitation volumes, leading to the situation that in some cases the non-cavitating part becomes dominating for blade rate. Furthermore, the designs are characterized by the fact that the higher harmonics do not necessarily decrease, but some of them can take comparable values to blade rate. This makes it difficult to decide whether a specific propeller design is acceptable or not when the decision is only based on the judgement of the pressure pulses, which are typically measured in appropriate testing facilities. Therefore, two major development needs are obvious: Firstly, methods are needed to evaluate propeller designs and their impact on the hull structure before a blade is tested. This requires a combination of numerical methods and correlation techniques. The propeller induced pressure fluctuations have to be calculated with sufficient accuracy, taking into account the effective wake on one hand as well as the interaction with the rudder. These loads, which have to be generated during the initial design stage before the contract is made have then to be transferred to an FEM model of the steel structure which then allows to optimize the steel design to fulfill contract specification values. The paper describes the essentials of these methods and strategies.

1 Problem

A prediction of propulsor induced pressure pulses is difficult because of the complex conditions within the flow in the field of the aftbody. The interactions between the wake field, the propulsor and the aftbody require a simulation of a fluid with high Reynolds numbers. The flow around a real ship has Reynolds numbers round about 10^9 . Even, one profile of a propeller blade is working at a Reynolds number round about 10^7 . Furthermore, cavitation is a problem which cannot be ignored. High Reynolds numbers cause in difficult turbulent flows, which are not easy to simulate with RANSE methods. Furthermore such calculations are very time expensive. For a prediction of pressure pulses within the project state such methods are not practical, yet. Methods of less numerical effort are of less physical accuracy. But it has to be analysed how big this error is and whether it is acceptable or not.

An analysis of modern propellers shows that the non-cavitating part is one main cause of the pressure pulses. Because of low cavitation volumes of such propeller designs, the cavitating part of propeller are not any longer the primary reason. This effect can be recognised by comparing cavitating and non-cavitating measurements of the same design. There are measurements where a design has pressure pulses of 2kPa in the non-cavitating case and only 2.2kPa in the cavitating case. This shows that it is first of all important to focus more on the non-cavitating effects and to handle afterwards the cavitating part of the pressure pulses. Therefore, this paper concentrates its focus to the non-cavitating part.

2 Numerical Methods

RANSE methods like *Ferziger and Peric (1996)* are modelling the physics on a high level and they generate much information of the three-dimensional flow, but they need extensive computation power. Moreover, the grid generation should not be underestimated. Simple methods like 'Momentum theory' or 'Lifting-line theory' are easy to use, but they do not model the geometry of a propeller with sufficient accuracy.

The Vortex-Lattice method is an approach in between. It is able to fit a two-dimensional distribution of lifting surfaces over the blade and downstream. Fig.1 shows such a propeller model. These lifting-surfaces are used to find the circulation Γ over the blade. A good overview of such methods are described in *Bertram (2000)*.

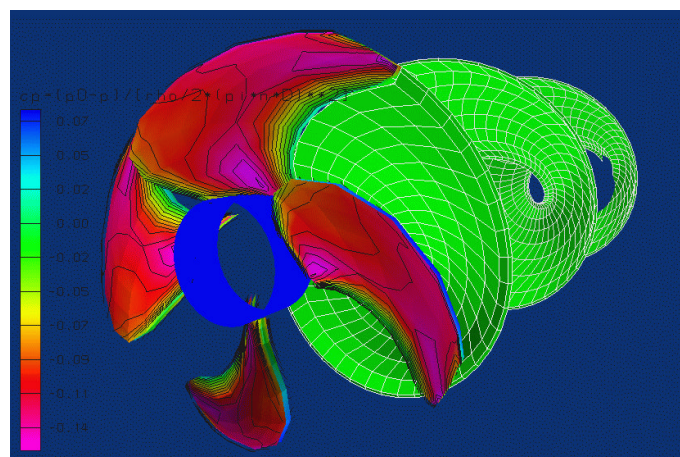


Fig.1: Propulsor modeled with a vortex-lattice method

The advantage of this method is that modern computer technology is able to handle the necessary numerical effort easily. A simulation of a propulsor at the ship's aftbody is finished after few minutes. Therefore, there is a possibility to do very many simulations and to do parameter studies. For these reasons the vortex-lattice method has been chosen within the ship design software E4 which has been used for these study.

2.1 Data Model

For the simulation models of the propulsor and the ship's aftbody are needed. A ship has been designed by a number of frames and longitudinals. Using this description a mesh of panels can be generated, which is used for calculation. As seen in Fig.2, there is only one regular grid, with $n \times m$ panels. The generation of these grids is an easy, fast and mostly automatic process, which is important for a fast design stage. The propulsor is defined with a set of discrete blade sections. For the vortex-lattice method a grid of $n \times m$ panels is generated automatically, too.

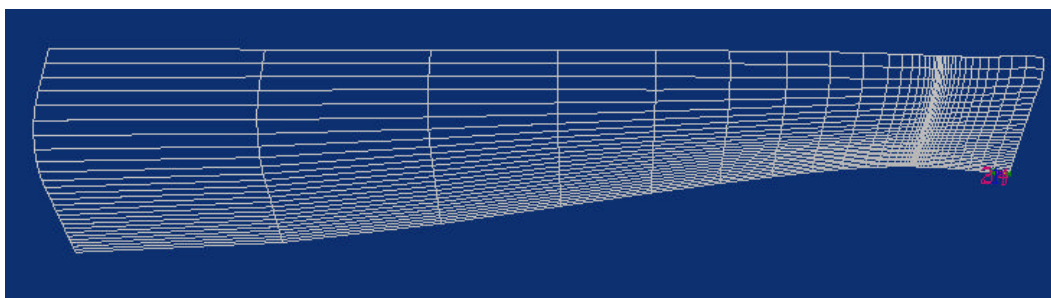


Fig.2: Calculation grid of an afterbody

The definition of the wake field is also important (see example Fig.3). If no wake field is available, simulations are possible with a homogeneous wake field. But this will downgrade the quality of simulation very much. Such a wake field is normally measured in a model testing tank environment. If such a measurement is not available, perhaps a model has not build already, a wake field of a comparison ship is better than nothing and will result a much better simulation than calculations with a homogeneous field (this will be explained later).

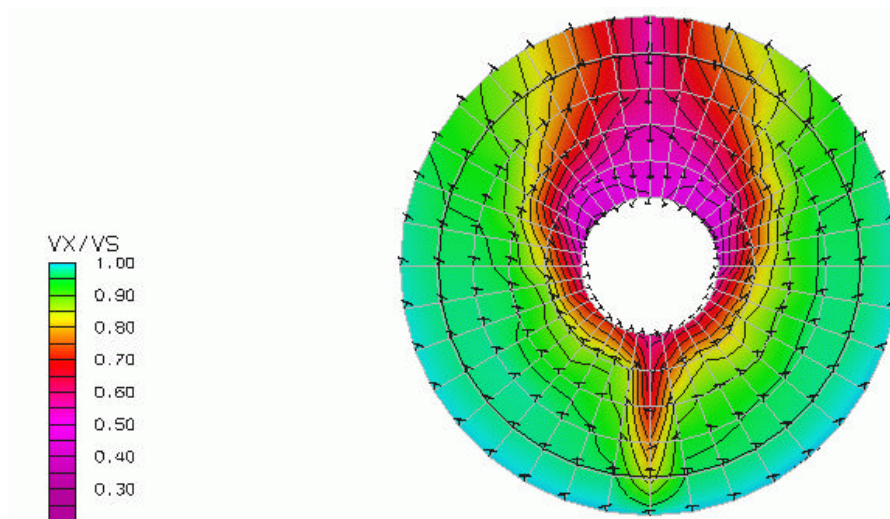


Fig.3: Measured wake field

2.2 Parameter studies of the simulation method

First of all the numerical stability has been tested. Therefore, different propellers have been simulated varying the position, revolution and grid resolutions. The results of these parameter studies are as follows:

- Position of the propulsor

The pressure pulse depends basically on the distance between propulsor and aftbody. Variations in the x or y direction are only important when the geometry of the hull differs in height:

$$P \sim \frac{1}{\Delta z^2} \quad (1)$$

This result is not surprising because it is consistent with theoretical considerations.

- Revolutions and inflow speed

Variations of revolution and speed of incoming flow are not resulting in a new issue, too. The simulation behaves as theoretical assumptions. If speed slows down or revolution turns up there is an increase of pressure pulses.

- Grid resolution

An interesting result is, that the vortex-lattice method is rather stable in reference to grid resolution. Changes in grid resolution near the pressure pulses maximum do not cause significant alterations of the result if the initial grid was generated in such a way that the area of the pressure pulses is no discretise too low. This critical is fulfilled if the neighbours of the cell with the maximum pressure pulse do not differ more than round about ten percentage. In this case a higher grid resolution will not change the result significantly.

As a conclusion of these parameter studies it can be highlighted, that the Vortex-Lattice method is a stable and easy to use tool for the simulation of pressure pulses. The grid generation can be handled with simple and mostly automatic algorithms. It is numerically stable and insensitive to variations of the grid resolution. Therefore, it is possible to use such a tool in the design stage of a ship, when engineers are dealing with rapidly changing design variations. In the next step it is necessary to analyse the quality of the simulations.

3 Simulation and Measurements of Non-Cavitating Propellers

The measurements of pressure pulses have to be compared with numerical simulation. Furthermore variations of wake field have been tested. For this work there has been a set of fourteen propellers. There are measurements of the non-cavitating part of the pressure pulses. These measurements have been done in the HYKAT of the HSVA (Hamburg Ship Model Basin).

In simulation three different wake fields have been used:

- Homogeneous wake field (SI1)
- Measured wake field (SI2)
- Measured wake field with a correction for the ruder influence and effective wake(SI3)

The correction of wake field SI3 have be done with potential theory methods, too. The acceleration of the propulsor has been added to the wake field in a previous simulation. In the same way the wake field deformation resulting from the rudder has been calculated. The results of simulation and measurement are listed in Table I.

Table I: Results of simulation and measurements

Propeller No.	SI1 [kPa]	SI2 [kPa]	SI3 [kPa]	Measurement [kPa]
1	1.20	1.76	1.83	2.80
2	1.03	1.47	1.50	1.60
3	1.37	2.08	2.18	2.70
4	0.89	1.28	1.34	1.40
5	1.06	1.67	1.73	1.50
6	1.15	1.74	1.83	2.20
7	1.06	1.61	1.70	2.20
8	1.04	1.55	1.63	2.20
9	0.87	1.22	1.25	1.26
10	0.57	0.76	0.77	0.82
11	0.81	1.63	1.64	1.35
12	0.80	1.67	1.68	2.04
13	1.56	2.08	2.15	2.13
14	1.32	1.74	1.80	1.62

Figs.A-1 to A-3 (Appendix) show the quality of the simulation. Fig.A-1 has been generated by simulating a homogeneous wake field. Every point is one propeller. The abscissa shows the simulation result and the ordinate denotes the measurement value. In the same way Fig.A-2 shows results of simulations of a measured wake field. The last simulation Fig.A-3 displays the calculations for the measured wake field with the correction for effective wake and rudder influence.

If the simulation would have been perfect, all points have lain on the line $y = f(x) = x$. In this case the simulation would have calculated the same value as the measurement. Because there are differences to the ideal case, the numerical errors have to be analysed. While comparing the data it becomes obvious that the simulation has the tendency to under-estimate pressure pulses. Most of all the homogeneous wake field shows this effect. This is not surprising because a homogeneous wake field has no gradient within direction of the rotating blade. Such a gradient induces timely changing fluid velocities and therefore a changing pressure. Consequently gradients in the rotation direction have to be an important aspect for the calculation of pressure pulses.

These theoretical assumptions correlate with the results of Figs.A-2 and A-3. By using the measured wake field there is a big improvement of simulation quality. The corrected wake field correction brings about an improvement, too. Therefore it is obvious that it is important to model the physics as accurate as possible. A comparison of different simulation types is shown in Fig.A-4 in the appendix.

4 Correlation between Simulation and Measurement

In the next step the error between simulation and measurement had to be analysed. The theoretical assumption is that the error can be split into two parts:

– Methodical Error

This is an error which affects the simulation every time in the same way. This type of error is becomes obvious clear in Fig.A-1. All simulation results are always too low. When this type of error becomes known, it is possible to generate a correction function to adjust the simulation.

– Random Error

This error is a pure random variable, where all effects which cannot be calculated are included. An indicator for this is, that in an average all errors sum up to zero.

4.1 Mathematical Model

The task now is to generate a function to minimize the methodical error. To do this some definitions become necessary:

$Y = \{y_1, \dots, y_n\}$	Set of all measurements
$y_i \in R^+$	Measurement of i-th propeller
$\hat{Y} = \{\hat{y}_1, \dots, \hat{y}_n\}$	Set of all corrected simulations
$\hat{y}_i \in R^+$	Corrected value of the i-th propeller-simulation
$X = \{\vec{x}_1, \dots, \vec{x}_n\}$	Set of all simulated propeller
\vec{x}_i	Set of calculations for the i-th propeller
$n \in N^+$	Number of tested propellers

\vec{x}_i is a vector because of the different types of simulation for one propeller (homogeneous, measured and rudder-corrected wake field, Fig.A-4). With these definitions it is possible to generate the following function:

$$\hat{y}_i = k(\vec{x}_i) \quad (2)$$

This function has to be evaluated. As in *Krickeberg and Ziezold (1977)*, a most common criteria is the quadratic error:

$$E(X, Y) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3)$$

For correlation of propeller data the following type of correlation function has been chosen:

$$k(\vec{x}_i) = \sum_{j=1}^m a_j \cdot f_j(\vec{x}_i) \quad (4)$$

In this approach the variable parameters are $a_{1..m}$. The functions $f_{1..m}$ are free. The only restriction is that they have to be independent from $a_{1..m}$. Otherwise, mathematical problems can appear. But this function is already powerful. The typical methodical errors are mostly offsets, linear and quadratic errors. Eventually logarithmic or exponential errors. All these functions can be handled. The task now is to find a set of values for vector $\vec{a} = \{a_1..a_m\}$ which minimize the error $E(\vec{a}, X, Y)$.

This is a typical optimization task. There are simple methods like the Newton algorithm, or more complex approaches like genetic algorithms or neural networks.

The advantage of approach in function (4) is the possibility to solve the optimisation analytically. This can be done because of independence of $f_{1..m}$ from vector \vec{a} . The minimum of Eq.(3) is reached when the derivative to all elements of \vec{a} is zero (Eq.(4) in Eq.(3) and then derivation to a_k):

$$\frac{\partial E}{\partial a_k} = \frac{1}{n} \sum_{i=1}^n 2 \cdot \left(y_i - \sum_{j=1}^m a_j \cdot f_j(\vec{x}_i) \right) \cdot (-f_k(\vec{x}_i)) = 0 \quad (5)$$

This is a linear system of equations, with m equations for m variables. In a next step these equations can be written in this form:

$$\sum_{i=1}^n (y_i - g(\vec{x}_i)) \cdot f_k(\vec{x}_i) = \sum_{j=1}^m a_j \cdot \sum_{i=1}^n f_j(\vec{x}_i) \cdot f_k(\vec{x}_i) \quad (6)$$

The following variables can be defined:

$$b_k = \sum_{i=1}^n y_i \cdot f_k(\vec{x}_i) \quad (7)$$

$$m_{kj} = \sum_{i=1}^n f_j(\vec{x}_i) \cdot f_k(\vec{x}_i) \quad (8)$$

Now it is possible to write the optimisation problem in a simple matrix:

$$\vec{b} = (b_1, b_2, \dots, b_m)^T \quad (9)$$

$$\vec{a} = (a_1, a_2, \dots, a_m)^T \quad (10)$$

$$M = \begin{pmatrix} m_{11} & \dots & m_{1j} & \dots & m_{1m} \\ \dots & \dots & \dots & \dots & \dots \\ m_{k1} & \dots & m_{kj} & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ m_{m1} & \dots & \dots & \dots & m_{mm} \end{pmatrix} \quad (11)$$

$$\vec{b} = M \cdot \vec{a} \quad (12)$$

Solving this system of equations results in a parameter-set for \vec{a} so that the quadratic error of function $k(\vec{x}_i)$ is minimal.

4.2 Test of Different Correlation-Functions

Next some correlations have been tested. The variables of the optimization algorithm have been defined in the following way:

y_i	Measurement from HSVA
x_{i1}	Simulation with homogeneous wake field
x_{i2}	Simulation with measured wake field
x_{i3}	Simulation with ruder corrected wake field

The mathematical optimization function is Eq.(3). For a better understanding the average percentage differences are listed in Table I. This error is defined as:

$$E_{perc.}(\vec{a}, X, Y) = \frac{100\%}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{\hat{y}_i} \quad (13)$$

Different functions for $k(\vec{x}_i)$ have been tested. Table I shows an overview of these functions and the resulting error. For completeness the error values of non-correlation are included, too.

Table I: Results of different correlations

Num.	$k(\vec{x}_i) =$	a_1	a_2	a_3	a_4	$E_{perc.}$
1	x_{i1}	-	-	-	-	76.0 %
2	$a_1 x_{i1}$	1.737	-	-	-	19.5 %
3	$a_1 x_{i1} + a_2$	1.456	0.312	-	-	20.5 %
4	$a_1 x_{i1}^2 + a_2 x_{i1}$	-0.433	2.242	-	-	20.3 %
5	$a_1 x_{i1}^2 + a_2 x_{i1} + a_3$	-1.863	5.448	-1.709	-	17.5 %
6	x_{i2}	-	-	-	-	20.2 %
7	$a_1 x_{i2}$	1.165	-	-	-	15.4 %
8	$a_1 x_{i2} + a_2$	1.285	-0.199	-	-	14.7 %
9	$a_1 x_{i2}^2 + a_2 x_{i2}$	0.084	1.022	-	-	14.4 %
10	$a_1 x_{i2}^2 + a_2 x_{i2} + a_3$	-0.0189	1.340	-0.236	-	14.8 %
11	x_{i3}	-	-	-	-	17.4 %
12	$a_1 x_{i3}$	1.127	-	-	-	14.7 %
13	$a_1 x_{i3} + a_2$	1.125	-0.213	-	-	14.3 %
14	$a_1 x_{i3}^2 + a_2 x_{i3}$	0.0904	0.966	-	-	13.8 %
15	$a_1 x_{i3}^2 + a_2 x_{i3} + a_3$	0.0326	1.152	-0.143	-	14.1 %
16	$a_1 x_{i1} + a_2 x_{i3}$	-0.199	1.254	-	-	14.1 %
17	$a_1 x_{i1} + a_2 x_{i3} + a_3$	-0.187	1.368	-0.209	-	14.0 %
18	$a_1 x_{i1} + a_2 x_{i3} + a_3 x_{i1} x_{i3}$	-0.424	1.268	0.114	-	13.6 %
19	$a_1 x_{i1} + a_2 x_{i3} + a_3 x_{i1} x_{i3} + a_4$	0.0602	1.450	-0.120	-0.388	14.4 %
20	$a_1 x_{i3} + a_2 x_{i1} x_{i3}$	1.087	0.0353	-	-	14.5 %
21	$a_1 x_{i3} + a_2 x_{i1} x_{i3} + a_3$	1.438	-0.0949	-0.350	-	14.3 %

Comparison between these function show that simulation with a homogeneous wake field need correlation most often. The error level reduces itself from 76% to 15.5%, Fig.A-5). Figs.A-6 to A-8 show the quality of different correlation functions.

The first three parts are correlations to the homogeneous, the measured and the corrected wake field. The last part (No. 16-21) is an effort to correlate information from simulations with different wake fields. But it is easy to see, that there is no big potential for such complex correlations.

The enhancement of quality by correlating a simulation with a homogeneous wake field, is at the first view very big. But at the second view it is obviously, that this type of simulations has disregard an important physical effect. The use of a measured wake field (No. 6-10) improves the simulation in one step round about so much as the best correlation of the homogeneous wake field.

The third part (No. 11-15) shows the best results. Which number of correlation should be used is not easy to answer. An easy to use linear approach has already a good quality. Approaches with

quadratic functions result in little bit better results. But the quadratic part of the correlation is very low. Therefore it is not easy to see, whether this quadratic part is real or if it is a random effect resulting from the low number of tested propeller.

5 Conclusion

As a result it should be emphasized that the effect of the wake field is important for the pressure pulses. A homogeneous wake field under-estimates the pressure pulses. All further correlations can not real adjust this error. Only if there is absolutely no useful wake field available, information still can be generated by a correlation based on simulation with a homogeneous wake field. The use of a wake field improves the quality of the simulation significantly. If physical effects like the effective wake field or rudder influences are available the numerical quality improves, too. Because these both effects are easy to calculate, they should be used. Afterwards a correlation can adjust the simulation.

In spite of this correlation the error level is still high. Therefore, further analysis is necessary to find the main effects which are responsible for the differences between simulation and measurement, namely the quality of the wake field, the distribution of circulation above the blade and influences of effects in the vicinity of the blade tip. Also the cavitating part of pressure pulses has to be analysed.

References

- BERTRAM, V. (2000), *Practical Ship Hydrodynamics*, Butterworth-Heinemann, ISBN 0-7506-4851-1
- FERZIGER, J.H.; PERIĆ, M. (1996), *Computational Methods for Fluid Dynamics*, Springer, ISBN 3-540-65373-2
- KRICKEBERG; ZIEZOLD (1977), *Stochastische Methoden*, Springer, ISBN 3-540-57792-0

Appendix: Graphics

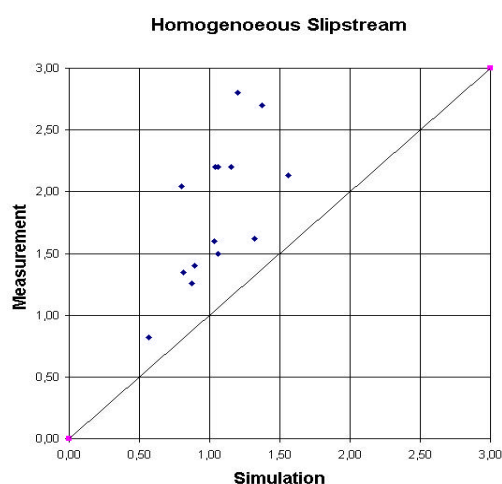


Fig.A-1: Simulation with homogeneous wake field versus measurement

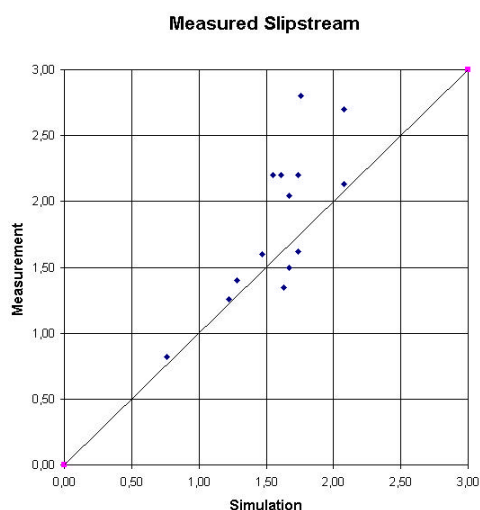


Fig.A-2: Simulation with measured wake field versus measurement

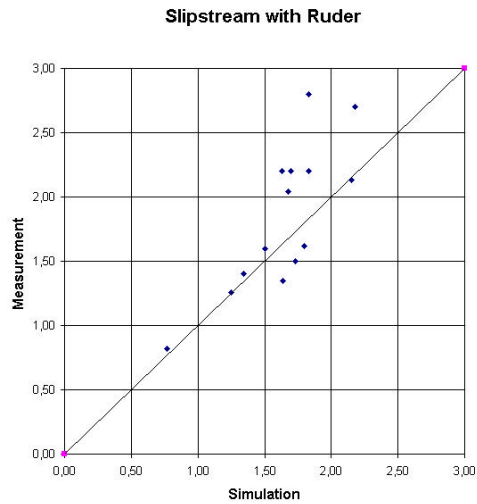


Fig.A-3: Simulation with wake field corrected for rudder versus measurement

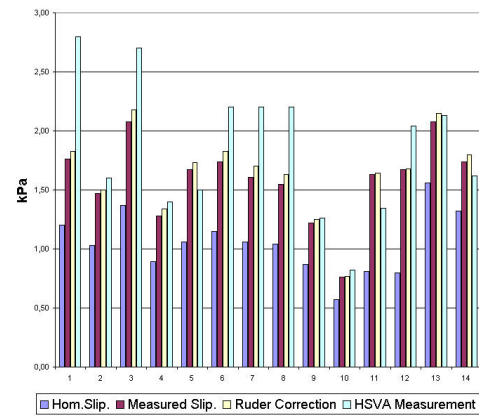


Fig.A-4: Comparison between the simulation types

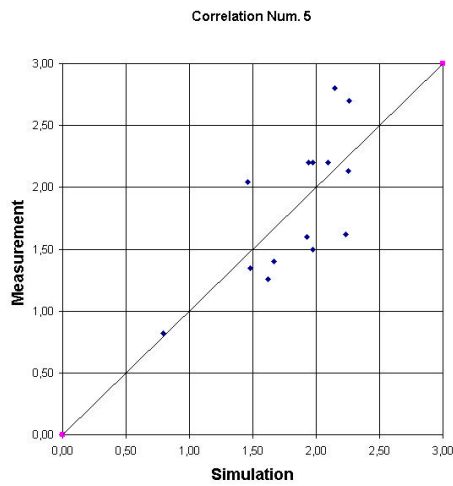


Fig.A-5: Quality of simulation results for function No.5

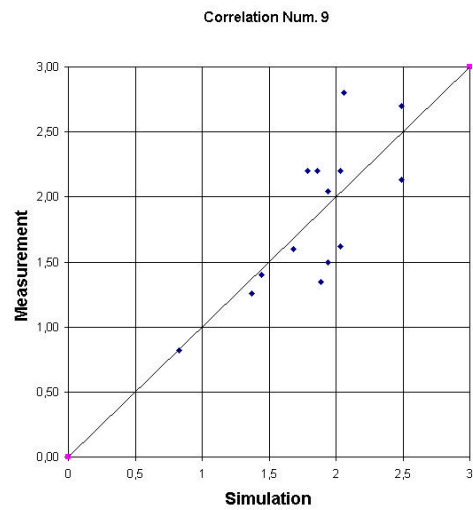


Fig.A-6: Quality of simulation results for function No.9

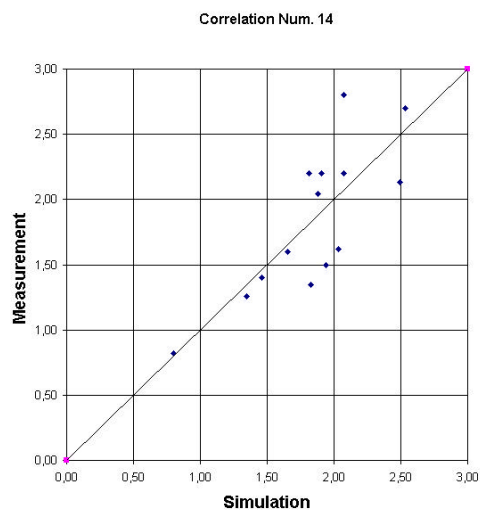


Fig.A-7: Quality of simulation results for function No.14

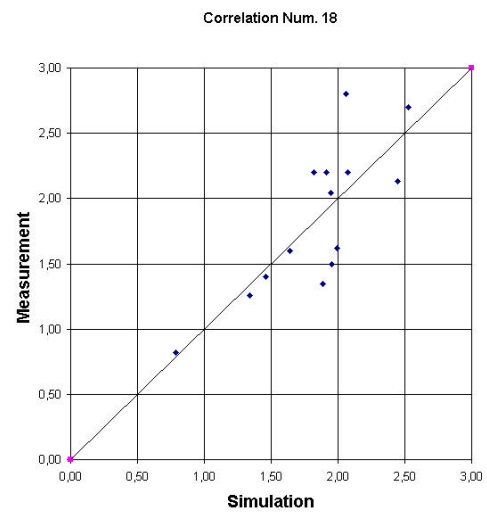


Fig.A-8: Quality of simulation results for function No.18

Index by authors

Abels	470	Mason, G.	391
Akinfiev	315	Massow	378
Ammeen	430	Matusiak	131
Andritsos	140	Millbourn	36
Armada	315	Montes	315
Aziz	154	Mutu	334
Bertram	5	Nabulsi	315
Blümel	73	Neef	363
Bohlmann	165	Nienhuis	190,263
Boonstra	223	Nieuwenhuis	190
Bortfeldt	419	Novitski	73
Boulougouris	175	Papanikolaou	175
Bronsart	111	Pedraza	315
Campana	322	Pereira	290
Coenen	263	Peri	322
Couser	391	Persson	61
Davidsson	61	Pietrzykowski	204
Dogliani	447	Pinto	322
Domeyer	42	Ponticelli	315
Enomoto	148	Pradillon	278
Estremera	315	Prieto	315
Faller	430	Queutey	300
Fernandez	315	Rodriguez	403
Ferrando	17	Ross	98
Fidani	81,140	Sahajpal	86
Fischer	122	Sarria	315
Fu	430	Sasaki	148
Furstenberg	252	Schmidt	154
Garcia	315	Schom	81,140
Gau	111	Schulten	236
Gehring	122	Schwill	42
Gongora	354	Seif	459
Gonzalez	315	Siksne-Pedersen	378
Grimmelius	236	Simpson	252
Haack	52	Smith	391
Harries	27	Stapersma	236
Henesey	61	Stinstra	223
Hees, van	263	Strang	447
Hess	430	Sucharowski	111
Hinnenthal	27	Traverso	17
Hupe	154	Tronstad	403
Ionas	334	Turan	340
Jahanbaksh	459	Türkmen	340
Koelman	223	Uriasz	204
Konsky	391	Vassalos	447
Krüger	52,470	Veelo	212
Langbecker	154,290	Vijver	223
Le Page	81	Visonneau	300
Lieburg, van	363	Weiss	81,140
Lommi	17	Weychardt	165
Luckau	111	Wuttke	42
Mäesalu	131	Yamato	148
Mason, A.	391		

4th International Conference on
Computer Applications and Information Technology in the Maritime Industries

COMPIT'05

Haus Rissen, Hamburg/Germany, 8-11 May 2005

Topics: CAD / CAM / CIM / Simulations / Virtual Prototyping / Virtual Reality
Robotics / Computer Aided Planning
Information Technology Standards/ Electronic Data Exchange/ Net Technology
Management Information Systems / Executive Information Systems
Artificial Intelligence
Management/Legal/Economical Aspects of Information Technology
In Shipbuilding, Offshore Engineering, Port and Ship Operation

Organisers: Volker Bertram, Heinrich Söding

Advisory Committee: (further members to be announced)

Manuel Armada	IAI-CSIC, Spain	Lawrence Henesey	Blekinge Inst. Techn., Sweden
Berend Bohlmann	FSG, Germany	Yvonne Masakowski	NWC, USA
Alain-André Clouet	Chantiers de l'Atlantique, France	Ehsan Mesbahi	University Newcastle, UK
Ludwig Furstenberg	ITE, South Africa	Ubaldo Nienhuis	TU Delft, NL
Martha Grabowski	Le Moyne College, USA	Marcos Salas	University Austral, Chile

Venue: The conference will be held at Haus Rissen, a conference venue in the picturesque outskirts of Hamburg-Blankenese, near the Elbe river promenade. Accommodation is available at the conference venue.

Format: Papers to the above topics are invited and will be selected by a selection committee. There will be hard-cover proceedings and papers may have up to 15 pages

Important dates:	30.09.2004	optional "early warning" of intent to submit paper
	12.01.2005	abstract received deadline
	15.01.2005	notification of acceptance
	07.03.2005	payment due for authors
	22.03.2005	final papers due

Fees: 450 Euro participation fee

Information: volker.bertram@ensieta.fr
www.ssi.tu-harburg.de/compit05.html

Sponsors: to be found...