23rd Conference on

Computer and IT Applications in the Maritime Industries

COMPIT'25

Pontignano, 7-8 October 2025

23rd International Conference on

Computer and IT Applications in the Maritime Industries

COMPIT'25

Pontignano, 7-8 October 2025

Edited by Volker Bertram

23rd International Conference on Computer and IT Applications in the Maritime Industries, Pontignano, 7-8 October 2025

https://doi.org/10.5281/zenodo.17175694

Download from: www.compit.info or https://zenodo.org/communities/compit/

Sponsored by

tutech.de

www.prostep.com

https://ocius.com.au

https://engys.com

www.wartsila.com

www.siemens.com

www.nagelld.no

www.numeca.de

Media Partners

https://hansa-online.de/

https://www.rina.org.uk/

Autonomy x 2 – Sustainably Powered Smart Unmanned Surface Vessels

Robert Dane, Ocius Technology, Randwick/Australia, robert@ocius.com.au
Nick Rozenauers, Ocius Technology, Randwick/Australia, nick.rozenauers@ocius.com.au
Ian Milliner, Ocius Technology, Randwick/Australia, ian.milliner@ocius.com.au

Abstract

This paper describes the concept of Ocius' Uncrewed Surface Vessels (USVs), the 'Bluebottles'. The drones are autonomous robots powered by solar, wind and wave energy making them also autonomous in terms of energy, allowing long periods of operation covering vast areas. The drones are team capable and approved by AMSA for operating in Australia's EEZ to conduct long-duration autonomous surveillance missions. Ocius is currently deploying Bluebottles for defence, oil & gas, and oceanography missions around Australia.

1. Our story – The long enjoyable road towards the "Bluebottles"

The Bluebottle range of USVs (Unmanned Surface Vehicles) are innovative unmanned drones designed to provide long-term autonomous surveillance and communications for defence, offshore, or oceanographic applications, Fig.1, https://ocius.com.au/. They represent a unique combination of technologies making them "doubly autonomous":

- The drones are "intelligence autonomous" in some decision-making, e.g. using Artificial Intelligence for autonomous collision avoidance and team capability. This is the normal autonomy that people have in mind when talking about autonomous underwater vehicles or autonomous (future) surface ships, e.g. *Bertram (2016).*
- The drones are also "energy autonomous" using wind, solar and wave power, with rechargeable batteries to ensure propulsion and manoeuvring at all times. Sustainably powered USVs like the Bluebottle drones could stay at sea indefinitely, in theory. A current practical limitation can be accumulating biofouling. Classical biocidal antifouling solutions have typical life cycles of 5 years before requiring replacement, but some of the evolving innovative solutions for biofouling may overcome the issue with biofouling limiting deployment times.

Fig.1: Bluebottle drones at sea

The Bluebottle solution is unique in its combination of intelligence and energy autonomy. It is the product of an evolution that spans almost 30 years (of trials and tribulations, as well as triumphs):

- 1997-2000
  - Winner of Advanced Technology Boat Race in Canberra with a boat with solar panels, Fig.2, that could angle to the sun and wind and fold away.
  - Formation of SolarSailor Holdings Limited.

- 2001-2014
  - Built solar ferry for Sydney Olympics, Fig.3, which won 2001 Australian Design Award of the year
  - Six hybrid ferries sold to Australia, China and Hong Kong carrying 10s of thousands of passengers

- 2015-2021
  - Enquiry from USA for building a platform that could go to sea 'forever'
  - Three successive defence innovation contracts
  - Seven USVs built and completed successful missions with Australian Border Force, RAN, Army, Ops Sovereign Borders and Marine Parks Australia, Fig.4

- 2022-Current
  - Contract for five Bluebottles for $5M with Warfare Innovation Navy branch (WIN branch).
  - Royal Australian Navy (RAN) operations contract
  - JAMSTEC contract, mapping underwater volcanoes off Japan
  - Marine Parks Australia
  - Oil and Gas trial
  - Trials for Royal New Zealand Navy, Fig.5

Fig.2: Boat Race winning solar-sail boat

Fig.3: SolarSailor, *Dane (2006)*

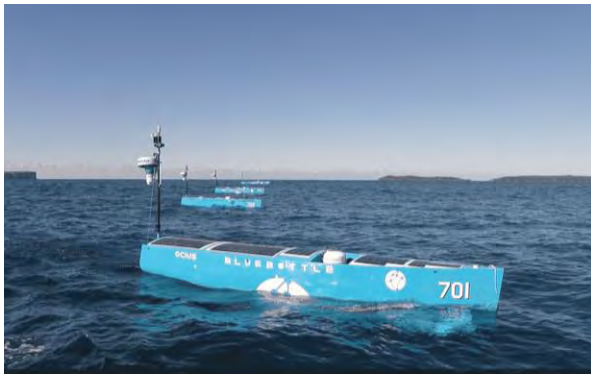


Fig.4: Bluebottle for Royal Australian Navy

Fig.5: Bluebottle trials in New Zealand

2. Bluebottle technology

Ocius' Bluebottle drones operate on energy harvested at sea. Thus, the 'Beth' type Bluebottle requires no fuel and no crew, hence no supplies. The 'Bathy' type Bluebottle offer enhanced power options at night and wind-still conditions using diesel hybrid propulsion. See Appendix I for technical specifications. Both Bluebottle types are self-deploying and self-retrieving and can roam widely or be kept on station virtually indefinitely. The following will focus on the 'Beth' type.

2.1. Persistent long-range ISR (Intelligence, Surveillance and Reconnaissance)

The ultimate objective of the development of the Bluebottle drones was to showcase an affordable, persistent, long-range detection capability, primarily to bolster the ability of navies and border forces to identify and monitor potential threats (e.g. submarines) from practical distances.

The "persistent" part required automatically sustainable energy supply with rechargeable batteries for power supply when the mix of solar, wind, and wave energy was insufficient. As illustrative example, Bluebottle drones off the coast of Australia operated for 35000 nm over two years, on a long-term ISR mission, Fig.6.

The "long-range detection" is based on an array of low-cost, low-weight sensors. Mission-specific sensor suites can be installed in the modular concept, fitted inside the hull and on the aft communications mast. An integrated and networked communication system supports live tracking and monitoring of the drones. They may be operating autonomously or under remote control.

Fig.6: Long-term Bluebottle deployment for ISR missions in Australia and New Zealand

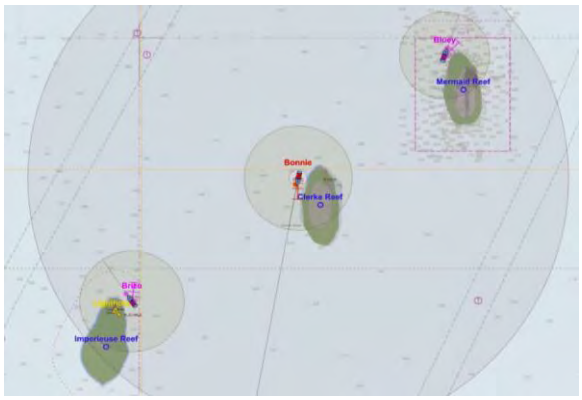


Fig.7: Bluebottles operating in a team

The drones have some "thinking" and decision autonomy including team capabilities, Fig.7, where Bluebottle drones communicate in a team and e.g. reconfigure location, selecting one vessel in a team to inspect a suspicious object, while the others continue to patrol a given area, etc.

The Bluebottle capabilities allow a variety of missions:

- Maritime Domain Awareness - Proven ISR capability with Australian Border Force, Regional Force Surveillance Group, and AMP (???).
- Enhanced ISR - Improved Radar, cameras, plus acoustic sensors and better Artificial Intelligence
- Anti-Submarine Warfare – Bluebottle equipped with a winch for a thin line array that detects and locates underwater vessels
- Communication Gateway - Bluebottle used as platform to connect with underwater assets
- Electronic Warfare - Monitoring of electromagnetic signals and spoofing
- Mine Counter Measures - Demonstrated with DST (https://www.dst.defence.gov.au/) Autonomous Underwater Vehicles

- Bathymetry - Single and multi-beam echo sounders can be fitted to Bluebottles to map underwater topography, Fig.8
- Offshore oil & gas/windfarms - Environmental monitoring, security and wildlife surveillance, Fig.9

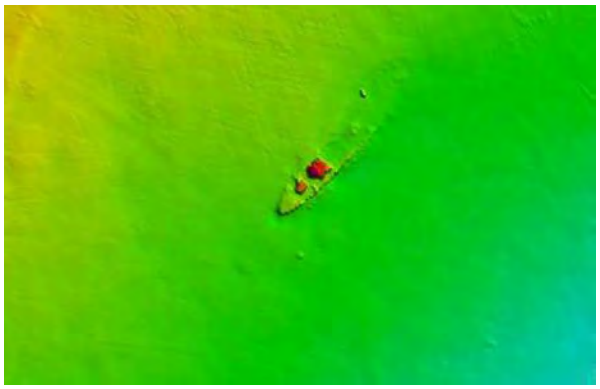


Fig.8: Bathymetric detection of wreck on seabed

Fig.9: Offshore underwater inspection

2.2. Interoperability

Autonomous USVs like the Bluebottles can and should be part of a larger (defence) eco-system, collaborating with other Bluebottles, other robotic systems (e.g. underwater drones), or manned vessels. Especially for countries with long coasts and sparse populations/harbours, Bluebottles can act as "satellites of the sea" with permanent surveillance of territorial waters. The combination of autonomous eyes on the front together with rapid response forces of manned vessels is an effective and also cost-effective approach to protection of security and economic interests for coastal states, as proven by the Australian Border Force, Fig.10.

Fig.10: Typical application cases for Bluebottles in ABF service

2.3. Deployability

The Bluebottle is easily deployable, using a choice of options, Fig.11:

Fig.11: Launch from a boat ramp

Fig.12: Launch by crane from ship

- Launch and recovery from a boat ramp, transport with road trailer, Fig.11
- Launch by helicopter from a ship
- Launch by crane from a ship, Fig.12

Only an area of 900 m^2 of hardstand is required. The containerised logistics support makes the Bluebottle easily deployable and operational.

2.4. Management

Support USV operations requires a skilled workforce of pilots and technicians. Ocius has supported customers with a flexible approach to vessel management, depending on their ability to support such a workforce. For customers who wish to own the platform, Ocius has sold boat and can support their operations with training, spares and maintenance. However, noting the personnel problems that navies are experiencing, Ocius developed an operational 'capability as a service' model, Fig.13, where Ocius supports depot level maintenance and operations up to the operational area, where it then hands control to a uniformed member of the navy. This model minimizes the resources needed and skilled personnel by navies to support uncrewed operations. This model is similarly used by Ocius for commercial applications of the Bluebottle, where it provides vessel time or data as a service.

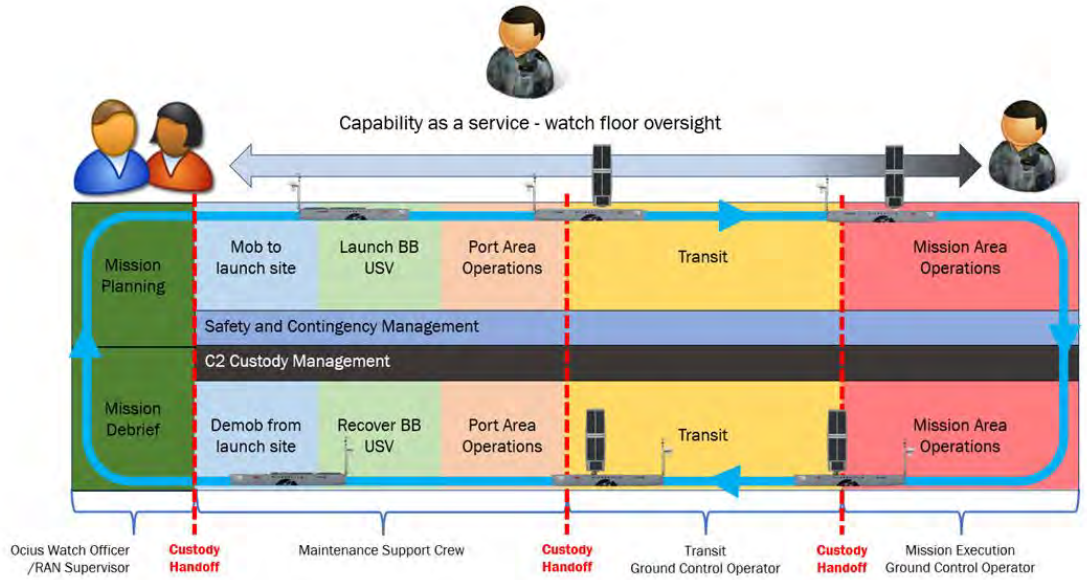


Fig.13: Example management style

3. Conclusions and outlook

The successful adoption of the Bluebottle USV by the Royal Australian and Royal New Zealand Navies for offshore surveillance validates the concept of USVs as force multipliers in border protection. However, the Bluebottle's impact extends far beyond military applications. Its long-range autonomy, low operating costs, and robust design make it ideally suited for growth industries like hydrographic surveys, environmental monitoring, and even anti-submarine warfare in a support role. Across these diverse sectors, operators are drawn to the Bluebottle's ability to minimize risk by using unmanned vessels, its alignment with environmental concerns through low emissions, and its cost effectiveness compared to traditional methods. The Bluebottle USV is not just a technological advancement; it is a disruptive force transforming the landscape of maritime operations.

Acknowledgements

We are grateful for the assistance of Volker Bertram in the writing of this paper.

References

BERTRAM, V. (2016), *Unmanned & Autonomous Shipping – A Technology Review*, 10th HIPER Conf., Cortona, pp.10-24, http://data.hiper-conf.info/Hiper2016_Cortona.pdf

DANE, R.; PARKER, G.; FAHR, M.; THOMSEN, C.; PATTEN, B. (2006), *Zero Particulate and Toxic Gas Emissions at the wharf by Commercial Hybrid-Electric Powered Vessels*, 5th HIPER Conf., Launceston, pp.447-454, http://data.hiper-conf.info/Hiper2006_Launcerston.pdf

Appendix I – Bluebottle technical specifications

	"Bathy"	"Beth"
Power	Solar, Wind, Wave, Diesel	Solar, Wind, Wave
Length	6.9 m	
Beam	1.3 m	
Draft	1.7 m	
Air draft – mast up	6.0 m	
Air draft – mast down	2.4 m	
Displacement	1500 kg	1100 kg
Top speed	6.0 kn	
Average speed	1.5 kn	3.0 kn
Sailing speed in 5 kn wind	1.4 kn	
Sailing speed in 15 kn wind	3.5 kn	
Sailing speed in 25 kn wind	5.5 kn*	
Operating sea state	5	
Max. sea state	7	
Wave powered speed	0.5-15 kn	
Energy storage	Battery 14-21 kWh Diesel 750 kWh	Battery 14-21 kWh
Solar power	Up to 1.5 kW	
Power allocated for payload	Average 0.85 kW for 30 days Max: 4 kW	Average 0.15 kW for 8 h sun Min: 35 W 10 days no sun Max: 2 kW
Payload	100 kg off sensor mounting bracket incl. MBES, sidescan sonar, sub-bottom profiler	150 kg dry in payload bay 150 kg wet in keel cassette module or winch
LARS	Boat ramp, or ship	
Keel winch diameter	n/a	1780 mm
Winch torque	n/a	120 Nm
Shipping	Two per 40' ISO shipping container	

* Depending on angle of attack

America's Cup Level CFD Digital Twins

Sven Albert, Numeca Ingenieurbüro, Altdorf/Germany, sven.albert@numeca.de
Thomas Hildebrandt, Numeca Ingenieurbüro, Altdorf/Germany, thomas.hildebrandt@numeca.de
Nathan Clero, Cadence, Brussels/Belgium, nclero@cadence.com

Abstract

This paper describes cutting-edge computational fluid dynamics (CFD) techniques and workflow procedures as applied for wind-powered vessels. While the focus of the illustrative applications lies on America's Cup yachts, the approach and employed software could be applied very similarly to assess and improve the performance of wind-assisted propulsion systems on cargo vessels.

1. Introduction

Designing sailing yachts for optimum performance is in essence not different from usual ship design optimization, but differs in the details due to the specific setting, materials used, dominant wind forces to consider, etc. The common design optimization key tasks are:

- Identifying the operational environment to design a vessel capable of meeting its challenges
- Optimizing vessel's design parameters to achieve the optimum performance in a representative mix of operational/ambient conditions (e.g. top speed)

Quantum leaps in performance may be possible if we leave our traditional design mindset or search space, e.g. by changing to foiling sailing yachts or by using wind-assisted propulsion systems (WAPS) on cargo vessels. In such cases, designers have to abandon experience-based design methods, and employ first-principles methods, such as Computational Fluid Dynamics (CFD).

Two specific events have occupied most of the high-performance sailing world in the past 5 years:

- America's Cup (AC), https://en.wikipedia.org/wiki/America's_Cup
  In the Barcelona 2024 edition, all teams were preparing, running statistical analysis on weather and sea state forecasts, and designing their vessels for a specific weather window. For the race, CFD was key to designing the fastest and most reliable boat, relegating model testing to history's shelves. Building a strong simulator fed with accurate physics was also a huge advantage in the sailors' preparation.
- Vendee Globe, https://en.wikipedia.org/wiki/Vendée_Globe
  In this 3-months race, manoeuvring and changing or readjusting the sails require an exhausting amount of energy from the sailor. Reducing unnecessary adjustments by building a precise performance map impacts the performance in the long term. Design philosophy shifted focus on attainable average speeds through better handling of the waves in the Southern Ocean.

Although the two races differ significantly in format (one a "sprint", the other a "marathon"), they both are built on the same two fundamental aspects:

(1) a design philosophy taking more and more into account the unsteadiness of real-life ambient conditions,
(2) and a consistent use and trust in CFD simulations as part of the design process.

The CFD tool of choice has been Cadence's Fidelity Fine Marine, https://www.numeca.de/en/products-cfd-solutions/, Fig.1. Its participation in the America's Cup spans over four editions where it has consistently found its way to the main event including three consecutive wins.

Fig.1: America's Cup yacht design using Fidelity Fine Marine, source: Emirates Team New Zealand

Fidelity Fine Marine is CFD software tailored for marine applications. It can be considered as a virtual towing tank. By solving the Reynolds-Averaged Navier Stokes equations (RANSE) for incompressible flows in a Volume of Fluid (VoF) formulation, it aims at modeling the free-surface flows, solving for the body motions in 6 Degrees of Freedom. The design of competitive sailing yachts requires almost the entire range of capabilities available in Fidelity Fine Marine, Fig.2.

Fig.2: CFD capabilities required for America's Cup yacht design

2. CFD in sailing yacht (and wind-powered ship) design

How is CFD used in such design projects? We could summarize this through three aspects:

- Understanding and formalizing the conditions in which the boat operates: wind and sea state statistics for intended operational area, over which time period (40 min race, several days, a few months, lifetime of vessel) and how passively/actively the systems can be operated (with associated models for the controller).
- Hull (and WAPS) design: finding a good or the best compromise between the technical constraints (stability, structural strength, class rules) and performance goals.
- Performance studies: Using CFD to map the ship's performance in a representative matrix of operational and ambient conditions as a base for decisions in design, but possibly also later in voyage optimization.

In the following, we will discuss the features required to achieve these goals.

2.1. Meshing

Meshing the geometry accurately is the foundation of any good CFD simulation. Amongst other critical requirements are an accurate representation of the geometry features, a high-quality boundary layer mesh, and a smart volume mesh, allowing a precise capturing of the free-surface and the pressure systems while keeping the cell count under control, Fig.3.

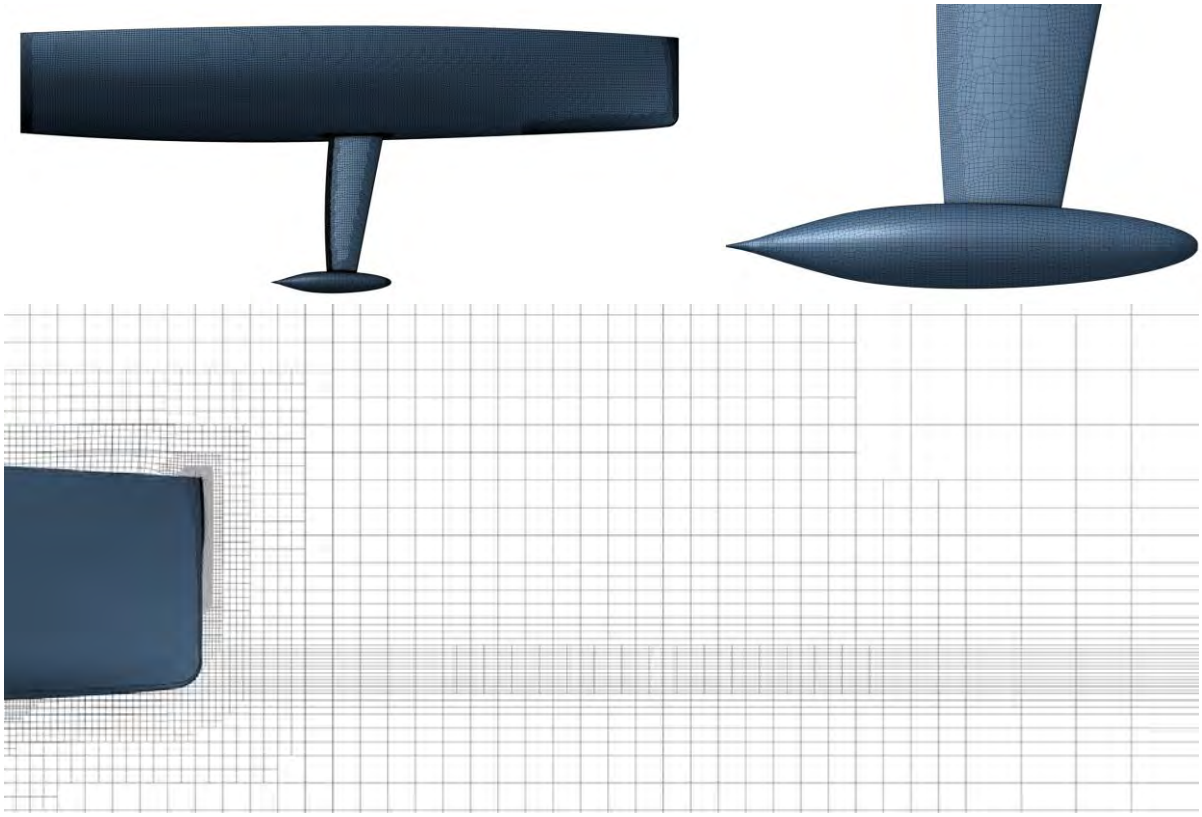


Fig.3: Typical mesh for sailing yacht CFD analysis

The boat motions can only be handled if the mesh can follow them. While the weighted mesh deformation technique is often used in marine CFD applications, sailing vessels can often reach large heel angles, beyond mesh deformability. A powerful alternative is to use the overset mesh approach, where the vessel and its background are meshed separately in different domains, Fig.4 (left). The meshes freely overlap, allowing any relative motion amplitude. The solver then interpolates the solution in between the domains to ensure continuity of the numerical solution. This technique can also be used to for appendages' rotation, like a rudder, opening a wide range of applications. Overset meshing also improves dramatically the mesh quality of hydrofoils, Fig.4 (right), *Robin et al. (2022)*.

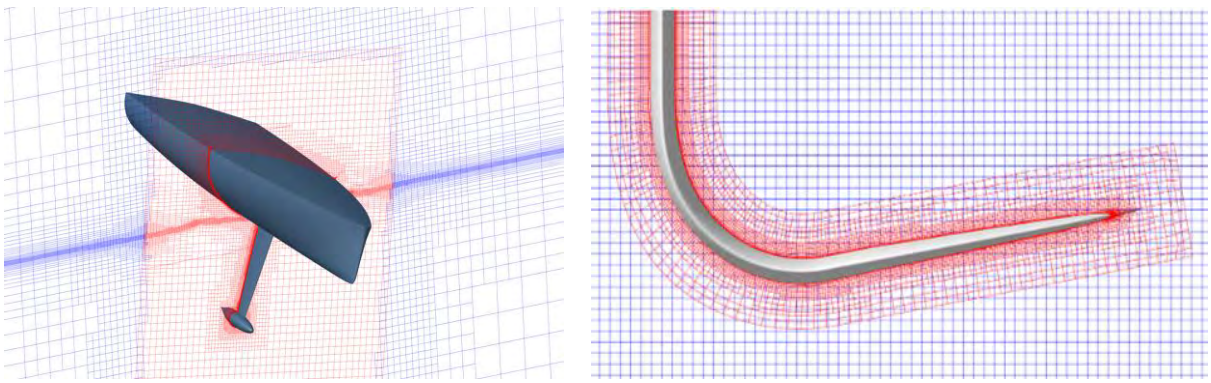


Fig.4: Overset mesh technology

2.2. Solving

The basics of hydrodynamic simulations in the marine world is resistance calculation, where typically we model the vessel with 2 degrees of freedom (free to sink and trim), but for yachts with 3 (free to heel as well). The VoF approach allows capturing free-surface deformation including breaking waves, Fig.5. CFD simulations use initial condition at rest and then accelerate to steady speed, similar to model tank tests. Several numerical methods can be used to accelerate the convergence time to quasi-steady state.

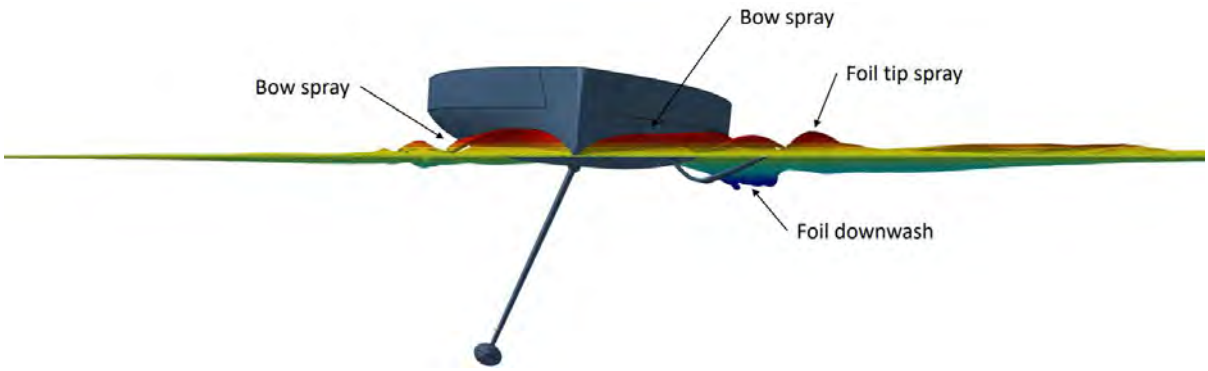


Fig.5: Free-surface deformation for yacht in CFD simulation using VoF approach, source: finot-conq

Designing for realistic ambient conditions required moving from resistance calculations to seakeeping simulations. Fidelity Fine Marine provides regular and irregular waves generation. Standard spectra (ITTC, JONSWAP, JONSWAP 3, and Pierson-Moskowitz) or a user-defined spectrum can be selected to generate a specific sea state.

CFD seakeeping simulations have been too expensive for many industry applications in the past. Capturing accurately the moving free surface in anticipation required a large number of cells through the entire domain. Fortunately, the Adaptive Grid Refinement (AGR), *Wackers et al. (2022)*, allows dynamic reconfiguration of cells during the simulation, reducing cell counts and computational time dramatically in seakeeping calculations, *Abgrall (2024)*. Cavitation and ventilation can also be predicted accurately using AGR, Fig.7.

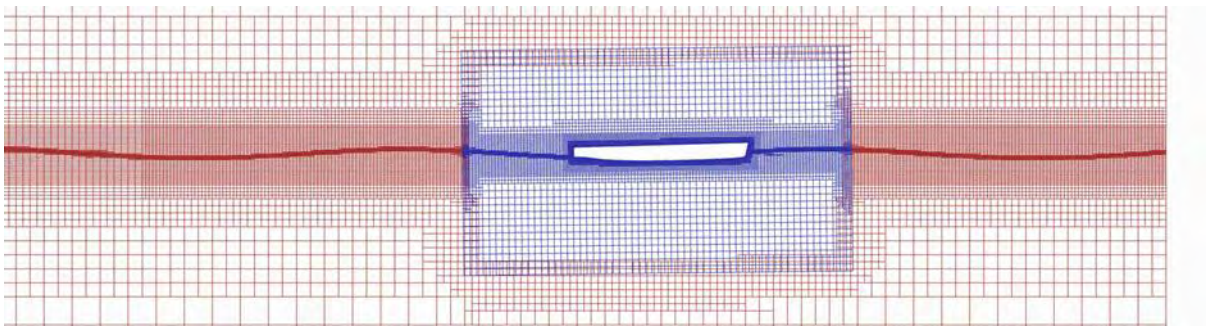


Fig.6: AGR mesh for seakeeping

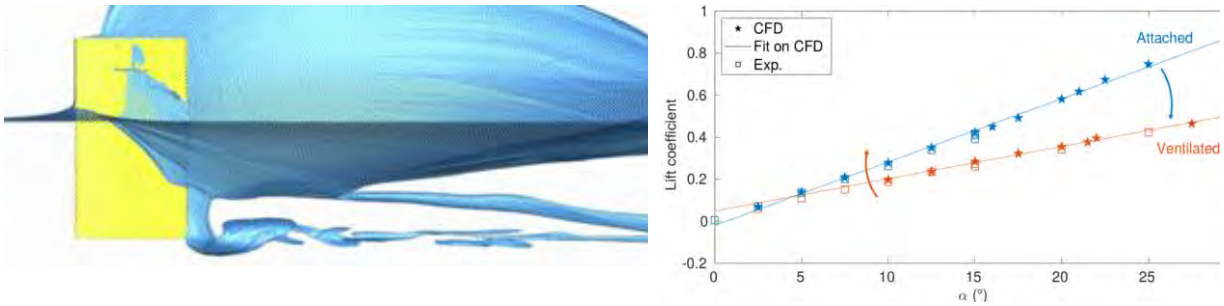


Fig.7: CFD simulation of ventilation at foil (left) and resulting lift compared to experiments (right)

High performance sailing also requires understanding fluid-structure interaction (FSI). The modal approach module in Fidelity Fine Marine provides such capability. After providing the modal structure file for the desired number of modes, the structure deformation can be resolved, Fig.8. While a controlled steady deformation can improve a design's efficiency by putting the geometry in a more efficient configuration at a given operating point, a structure too susceptible to fluttering can start oscillating out of control and be utterly destructive, Fig.9.

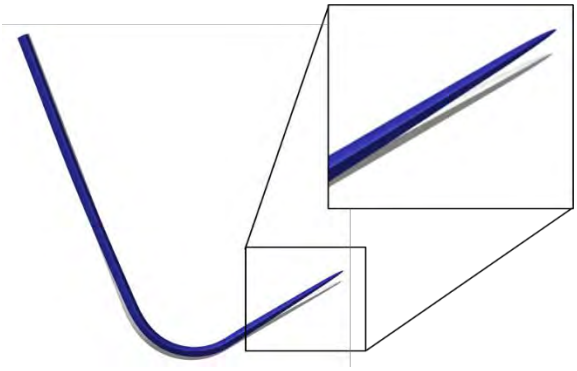


Fig.8: FSI result for foil on AC yacht

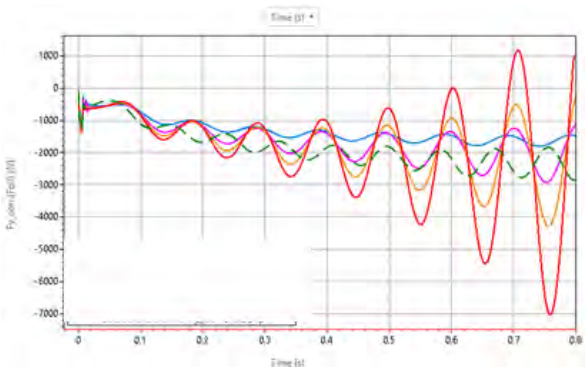


Fig.9: Stable and unstable fluttering

2.3. Workflow

One key element of a CFD chain is repeatability and consistency. This gives trust in the obtained results and allows multiplying the simulations with a limited amount of engineering time, removing at the same time the risk of human error.

The C-Wizard has been instrumental in bringing that consistency and automation in the design process. This tool prepares the entire setup for a given list of applications, applying state-of-the-art guidelines for the mesh and simulation setup steps of the CFD chain. Taking only naval-architectural information and conditions as input, the C-Wizard creates in a matter seconds to minutes the entire CFD project, Fig.10, increasing productivity while reducing potential errors. Several of the applications are key to sailing yacht design: resistance, seakeeping, position matrices for hull and foils, center of gravity and mass matrices, cavitation and transition setups for hydrofoils.

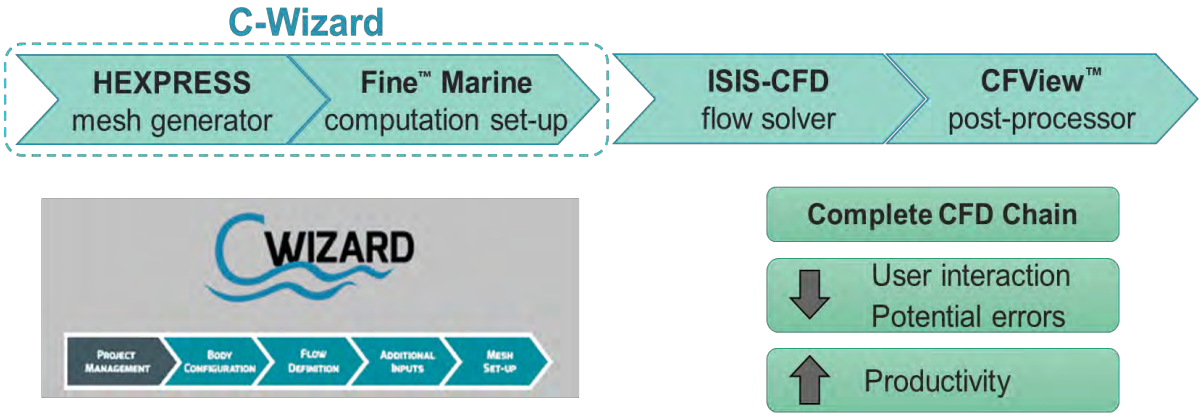


Fig.10: Workflow in C-Wizard

The C-Wizard can be run in matrix mode to be used for instance to feed a Velocity Prediction Program (VPP). This allows creating very robustly a large number of simulations varying the position of the boat in a highly consistent way across a pool of geometry variations. In the Sailing Yacht Research Foundation (SYRF) project, Fidelity Fine Marine was instrumented to run an end-to-end automated chain to simulate and analyze a matrix of 150 simulations on 3 geometries. To generate the 150 simulation setups, ready to mesh and simulate, the C-Wizard took only a few minutes.

Sailing yacht designers have always run Velocity Prediction Programs (VPP) to map the boat performances. The standard VPP approach consists in building a large matrix of cases and create a surrogate model to be able to interpolate between configurations. When building the hydrodynamic matrix, 80 to 200 CFD runs are required to have a surrogate model with sufficient quality to be representative, and this quality still depends on the quality of the chosen position samples. The finot-conq's Dynamic VPP does not prescribe the boat's attitude but instead solves the hydrodynamic position of the boat, propelling it using an integrated aerodynamic model, while running a boat speed optimization varying the sail power. As a result, for a given apparent wind angle/speed combination, in a single CFD run, the user retrieves the boat speed, the optimal sail power to reach it, and the solved position.

Since 2025, Fidelity Fine Marine offers the possibility to input directly an aero performance matrix representing any sail or wind assisted propulsion system, such as several sails, rigid sails, Flettner rotors, etc.

3. Commonalities and differences in CFD analyses for yachts and WAPS-powered ships

The discussion so far has been focussed on "America's Cup level" application to sailing yachts. Much of the experience in CFD simulations for sailing yachts can be applied towards WAPS-assisted ships, together with the extensive experience we have with Fidelity Fine Marine for the design and optimization of normal displacement ships.

The approach would differ only in a few aspects:

- The ratio between sail propulsion force and weight of the vessel is much smaller for WAPS-assisted. This justifies some simplifications. For example, heel may be neglected, and models may subsequently employ port-starboard symmetry with significant savings in computational effort.
- WAPS-assisted ships will always have rigid sails or rotor-sails. Fluid-structure interaction, ventilation and cavitation can be neglected completely.
- WAPS-assisted ships need a propeller model, just like regular cargo ships. A simplified propeller model using body forces (i.e. replacing the propeller by thrust and rotational forces in the cells where the propeller would be located) generally will suffice.
- Meshing depends on purpose of analysis and wind-assistance devices. Typically, one may use overset meshing strategies as described above. If the focus in on the wind-assistance device, e.g. a Flettner rotor, in an array of same devices, one may be modelled in high resolution, and the others in small resolution.

4. Outlook

In the future, design projects for sailing yachts and wind-assisted ships will benefit from current trends in development of the associated tools:

- Meshing
  - No more struggle with the CAD export formats, import seamlessly and clean the CAD directly in meshing tool.
  - Combination of meshing approaches, exploit the best of each approach into hybrid meshes.
  - Adaptive Grid Refinement will become usable for all hybrid meshes.
- Solving
  Mesh and solver improvement will progress hand in hand. The solver will adapt when new meshing technologies require a flow-solver adaptation, and the mesh requirements and guidelines will be adapted when improved numerical algorithms will call for it, ultimately reducing simulation times.

- Meta-modelling
  CFD simulations may be used to create fast meta-models. Systematic CFD simulations for applications with a handful of parameters can generate data for training machine-learning algorithms. Once trained, the resulting meta-model can give integral values (such as forces) and flow details (such as pressure and velocity fields) within seconds. The general procedure of CFD-trained meta-models has been successfully applied to propellers, *Albert et al. (2022),* and planing hulls, *Ahmed et al. (2023).* It could similarly be applied to e.g. a family of Flettner rotor designs.
- Optimization
  Libraries of meta-models for assorted sails and wind-propulsion devices may then be used in modular model generation, as e.g. in Hollenbach et al. (2020), and solving with sufficiently fast response times to apply formal optimization, both in design and operation, e.g. for dedicated wind-assisted ship voyage optimization.

References

ABGRALL, M. (2024), *Adaptive Grid Refinement Procedure for RANS-based Seakeeping Computations*, Master Thesis, ENSTA Bretagne, Brest, https://bu-catalogue.univ-brest.fr/cgi-bin/koha/opac-retrieve-file.pl?id=01c78f42235aa1b514c08e884f46140b

AHMED, O.; HARRIES, S.; LOHSE, J.; SALECKER, S.E. (2023), *Parametric Modeling, CFD Simulations, DoE and Machine Learning for the Design of a Planing Boat*, 22nd COMPIT Conf., Drübeck, pp.176-192, http://data.hiper-conf.info/compit2023_drubeck.pdf

ALBERT, S.; CORRÊA, R.; HILDEBRANDT, T.; DU TOIT, A.; KRENSKI, S. (2021), *Advanced CFD-based Design Approach for High-Performance Sailing Catamarans*, 13th HIPER Symp., Tullamore, pp.180-191, http://data.hiper-conf.info/Hiper2021_Tullamore.pdf

ALBERT, S.; HILDEBRANDT, T.; VAN DEN BOOGAARD, M.v.d.; ALESSI, G.; MALLOL, B.; WUNSCH, D.; CLERO, N.; AMADORI, D.; HIRSCH, C.; ZAMPIERI, L. (2022), *Using Machine Learning for Rapid Propeller Design Tools based on Numerical Series,* 14th HIPER Symp., Cortona, pp.34-42, http://data.hiper-conf.info/Hiper2022_Cortona.pdf

HILDEBRANDT, T.; ALBERT, S.; HARRIES, S.; REYER, M. (2017), *The Democratization of High Performance Marine CFD: A View from the Numerical, the Application and the Business Perspective*, 9th HIPER, Symp., Zevenwacht, pp.7-19, http://data.hiper-conf.info/Hiper2017_Zevenwacht.pdf

HOLLENBACH, U.; HANSEN, H.; HYMPENDAHL, O.; RECHE, M.; RUIZ CARRIO, E. (2020), *Wind Assisted Propulsion Systems as Key to Ultra Energy Efficient Ships*, 12th HIPER Conf., Cortona, http://data.hiper-conf.info/Hiper2020_Cortona.pdf, pp. 543-561

ROBIN, P.; LEROYER, A.; RICHEUX, J.; DE PRÉMOREL, D.; WACKERS, J. (2022), *Starting off the right foot with foil sock approach and AGR criterion,* 24th NuTTS Symp., Zagreb, https://cloudtowingtank.com/nutts-2022/

VAN DER KOLK, N.J.; KEUNING, J.A.; HUIJSMANS, R.H.M. (2019), *Part 1: Experimental validation of a RANS-CFD methodology for the hydrodynamics of wind-assisted ships operating at leeway angles*, Ocean Engineering 178, pp.375-387

WACKERS, J.; DENG, G.B.; RAYMOND, C.; GUILMINEAU, E.; LEROYER, A.; QUEUTEY, P.; VISONNEAU, M. (2022), *Adaptive grid refinement for ship resistance computations*, Ocean Eng. 250

3D Model-based Approval in Reality

Florin-Ciprian Luli, Vard Design & Engineering, Ålesund/Norway, ciprian.luli@vard.com
Myeong-Jo Son, NAPA, Helsinki/Finland, myeong-jo.son@napa.fi
Ole Christian Astrup, DNV, Oslo/Norway, ole.christian.astrup@dnv.com
Anders Madsen, DNV, Oslo/Norway, anders.madsen@dnv.com

Abstract

This paper explores the implementation of the model-based class approval (MBA) and verification scheme for new-build projects, replacing traditional 2D drawings with a 3D model. The study examines the results of applying the MBA process to a real-world design by VARD, involving the submission of a 3D OCX file exported from NAPA to DNV for approval. The review cycle, including class comments and feedback provided to the designer, is detailed. The paper highlights the differences between preparing traditional class drawings and the OCX model, focusing on time savings and the interaction between the designer and the construction yard. Additionally, the necessary adaptations to the design process are identified to maximise the benefits of the MBA.

1. Introduction

The use of 3D CAD is already widespread for design in shipbuilding, but the 3D model as a single source of truth - where the 3D model fully replaces 2D drawings across all processes - is not yet the norm. Current practices still rely on custom translators and point-to-point solutions. Most shipyards operate in a mixed mode, often described as "3D for design, 2D for production." Designers create detailed 3D ship models in modern CAD systems, yet 2D engineering drawings are still generated for downstream uses like fabrication, assembly, and class approval, *Huang et al. (2019)*.

This means the 3D model is not fully leveraged as the sole source of information, and the industry continues to depend heavily on drawings as deliverables. As a result, yards spend considerable effort converting and transcribing information from the model into drawings, a process that is both time-consuming and prone to inconsistencies. *Huang et al. (2019)* noted that this mixed 3D-and-2D process results in excessive data translation, information discrepancies and longer build cycles.

Over the last decade, a joint effort by the shipbuilding industry, *Astrup et al. (2022)*, has led to the emergence of a neutral and common specification for 3D models: the Open Class 3D Exchange standard (OCX). The OCX was developed to provide an interoperability framework that enables shipbuilders and classification societies to share digital workflows using a standardized 3D model specification. Importantly, the OCX addresses the needs of both shipbuilders and classification societies, supporting the fully digital exchange of structural information. Effectively, OCX acts as a conduit between the design tools and classification approval tools, highlighting the structural information required by the classification society and formatting it for efficient processing.

The APPROVED JIP, *Halfhide (2019)*, demonstrated the capabilities of the OCX, showing seamless information exchange between designer/yard and classification society, covering hull structure definition. To ensure its long-term adoption and development, the OCX Consortium, https://3docx.org, was established in 2021 to maintain, evolve, and promote the standard for the benefit of the shipbuilding industry.

This paper documents how VARD, DNV and NAPA applied model-based approval on a wind support vessel from VARD's 4 19 series, *NAPA (2025)*, using an OCX 3D model extracted directly from NAPA Designer. Instead of sending 2D drawings to the classification society, the OCX 3D model is submitted to DNV for review, Fig.1.

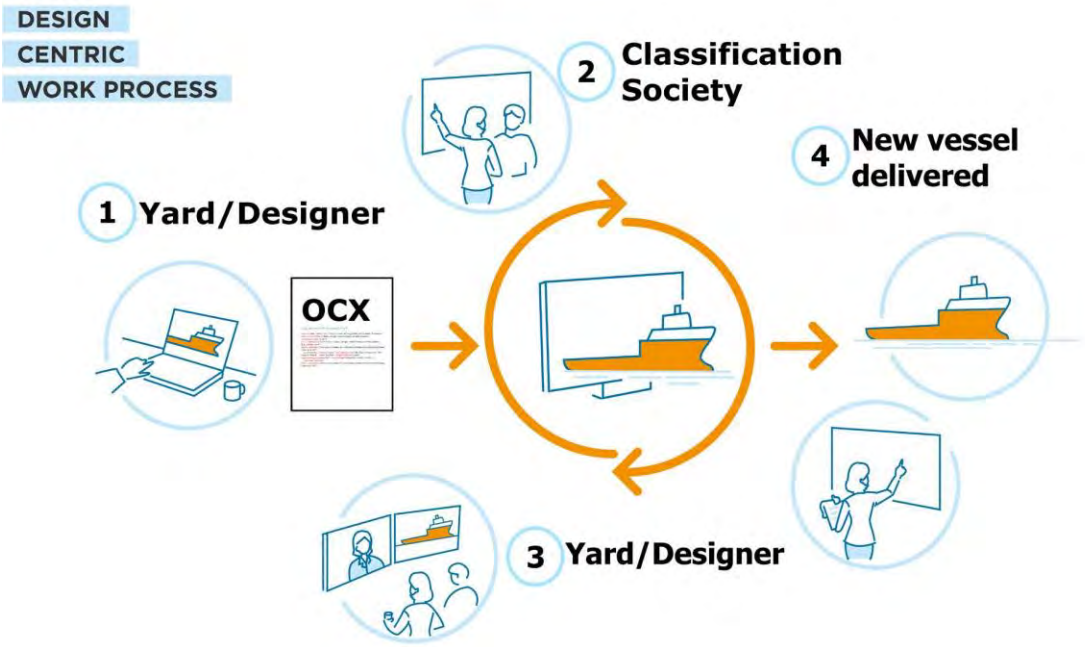


Fig.1: The 3D model information flow between VARD and DNV.

Sharing a common 3D model enables a fully digital and iterative work process (the numbers refer to the steps displayed in Fig.1):

1. VARD uploads the 3D design model to the classification society, similar to uploading any documentation.
2. DNV reviews the 3D model, performs rule calculations and provides comments and red markings directly on the design model, giving immediate feedback to the designer.
3. VARD makes the required design changes and engages in a model-centric dialogue with DNV. A new design revision is uploaded, documenting the changes.
4. At the end of the process, a new vessel is delivered with a shorter time to market, improved traceability, and quality.

2. One Single Source of Truth: One 3D Design Model

NAPA is a flexible, versatile, and open system with a 40-year history of development. It focuses on the upstream design process, including stability and naval architectural calculations, as well as structural design with a 3D model. The system's modernised user experience and efficient handling of a ship's entire 3D model have established it as the de facto standard tool for the early design phase, *Masui et al. (2023)*.

NAPA's "best-of-breed" approach allows it to interface with various essential tools, such as classification societies' rule calculation software, finite element model (FEM)-based direct strength analysis tools, design review and approval tools (for both 2D and 3D models), and downstream 3D CAD systems for production. This allows the NAPA 3D model to serve as a "One 3D design model" for a ship, *Masui et al. (2023)*.

NAPA has been involved in the APPROVED JIP, and, as a result, NAPA Designer was the first commercial product in the industry to offer standard export and import functionality for the OCX format. The concept of a "One 3D model" using OCX was previously demonstrated by *Son et al. (2022)*. For this concept to be practical, the key was the quality of the data transfer, the traceability between NAPA and other tools, and the ability to update an existing model by incorporating calculation results from classification societies and analysis results from FEM, as shown in Fig.2.

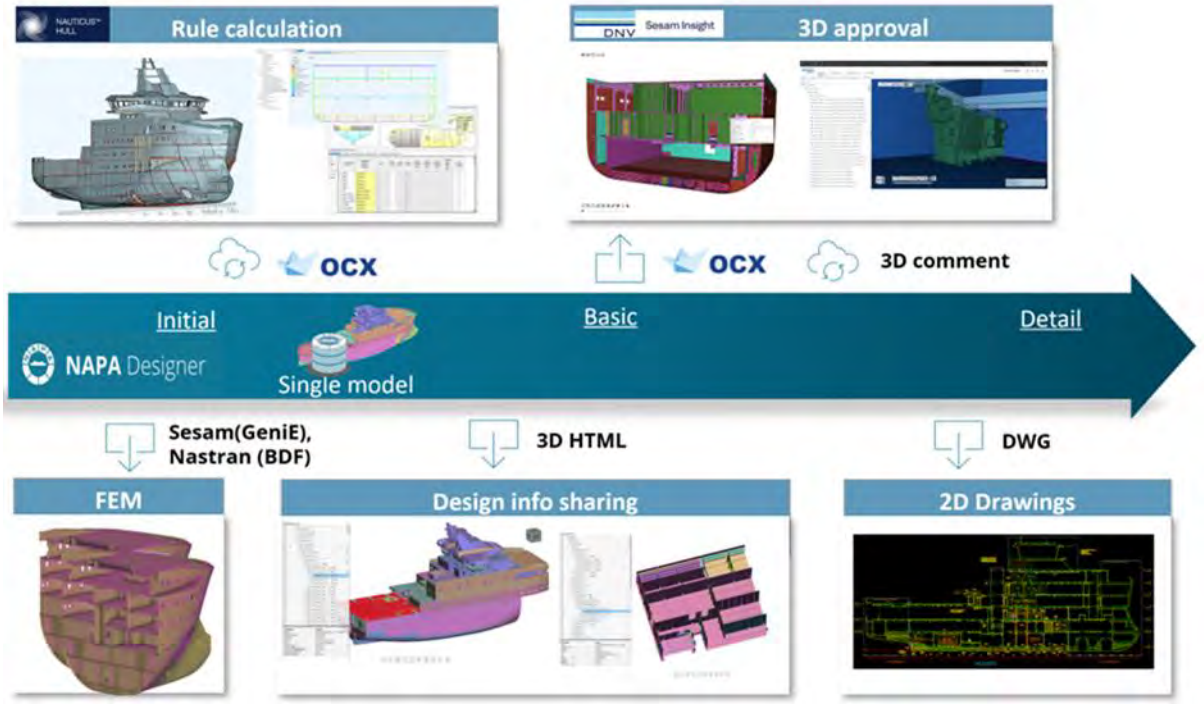


Fig.2: One 3D design model in NAPA for 3D approval

To address this, NAPA and DNV jointly developed a direct interface. This interface allows the DNV Nauticus Hull rule calculation tool to run as a rule engine API, providing live results directly on the target cross-section location of the 3D model in NAPA Designer. The core technology behind this functionality is the 3D model transfer facilitated by the OCX standard.

As the design matures with more inputs from the engineering calculations, FEM, and comments from the approval engineers of the classification societies, the 3D model can be shared with the relevant stakeholders as its latest version at any time during the design stage. This includes other disciplines, ship owners, and external partners. Sharing can be done using the OCX or NAPA Viewer (a web-based browser compatible with NAPA Steel 3D models), or the 3D HTML file, depending on the purpose of the sharing. The 3D HTML file, which is created based on the same technology as the 3D PDF, allows stakeholders to view and interact with the 3D workspace, including basic graphics interactions like clipping, selecting to check the properties or metadata of the model, and showing/ hiding objects, without needing a separate 3D viewer or license. In this research, we used the OCX format for rule calculations, 3D model review for approval and creating block-based models for site surveys. The 3D HTML file was used to share design information among different disciplines, effectively replacing 2D drawings.

Sharing the 3D model between DNV's approval system and NAPA Designer enables interactive collaboration by allowing the exchange of class comments directly on the 3D model. This includes creating interactive links to 3D model objects in NAPA Designer and restoring the exact view (camera, clipping) that was used when the comment was made. Additionally, a supplementary screen-shot from Sesam Insight (a web-based 3D model viewer of DNV) can be included with the essential comment text. Fig.3 shows how comments in a 3D model can be exchanged between DNV and NAPA. When these comments are imported into NAPA Designer, they appear in a comment window that is synchronised with NAPA Viewer. The comments can then be assigned to a specific design engineer and tracked using internal status indicators (open, in progress, review, closed). This ensures that the project coordinator can trace all open comments and communicate effectively with the classification society.

Fig.3: Comment exchange in 3D model between NAPA and DNV

3. One 3D Model: The Benefits

3.1. Model re-use - How VARD uses NAPA Designer to streamline model re-use in different software tools

Since implementing NAPA in our design processes, we have seen clear benefits – not only in faster modelling but also in the potential for many other applications that we have yet to fully explore. Our initial adoption of NAPA was somewhat slow, but once we discovered how customizable the software is, we began to recognize its broader potential.

For example, we already use our 3D model to generate DWG drawings, which are delivered to class societies for approval. This led us to ask whether the same 3D model could serve additional purposes. We began importing it into various other software tools to test compatibility and assess the benefits. Through these tests, we found that the NAPA model can easily be imported into NAUTICUS HULL (our dimensioning and scantling software) and into GENIE SOFTWARE (our FEM calculation tool). GENIE is mainly used for FEM analysis, reporting and sending information to classification societies.

Afterwards, our goal was to transfer the NAPA model into our detail design software AVEVA, ensuring all production information was accurately carried over between the two systems. The key requirement was that the transfer should be precise, without any loss or distortion of data.

From this, we realized the potential of having one single 3D model that could serve multiple purposes, supporting three different operations across three different software tools. This raised an important question for us: If we already have one model containing all the necessary information, could we use it as the single source of truth? Could we keep it continuously updated with the latest modifications and then use it across other processes?

We are now exploring how to adapt our workflows to make this possible. The idea is that the NAPA model becomes the foundation for all subsequent processes - a live 3D model that remains consistent and up to date throughout the design and production chain.

3.2. The Traditional Drawing-based Design Process by VARD

In this section, we want to highlight how NAPA shaped our process for extracting and delivering the drawing to the classification societies. Once the contractual General Arrangement is signed, we use it as the starting point for developing the 3D model.

In the first stage, we model all the bulkheads as dummy panels with a thickness of 2 mm, and we start to extract the first issue of the drawings, according to the document list required for submission to the classification society. Next, we add the girder layout, stiffeners and the large cutouts in the 3D model. This allows us to update the drawings with newly available information. In the following step, we focus on refining the 3D model by adding any missing structures, details, and end connections.

Once this update is complete, we regenerate the drawings and carry out a final quality check to ensure that no information is missing before delivering the package to the classification society, Fig.4.

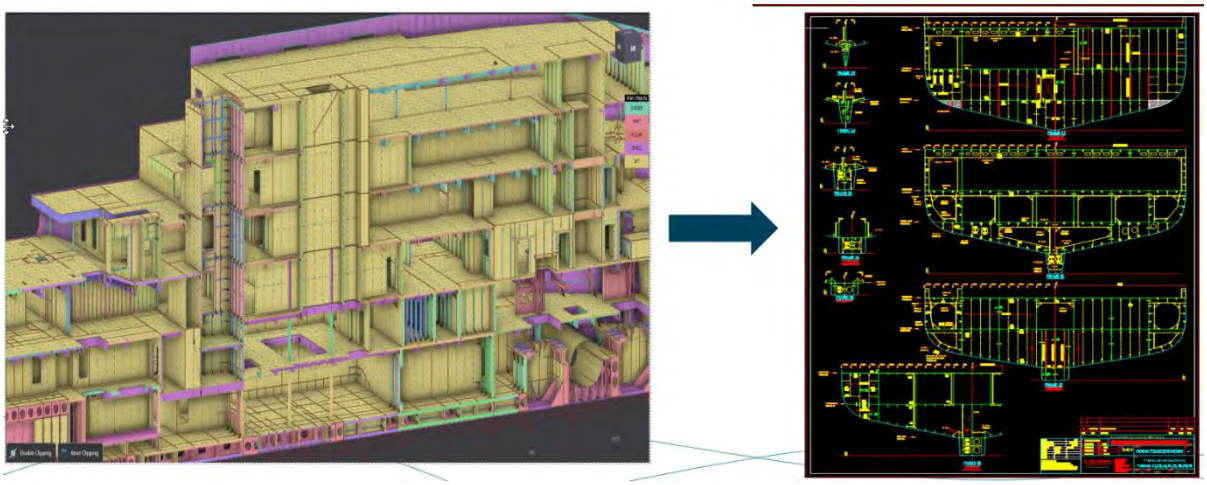


Fig.4: Extracting 2D drawings from the 3D model

We observed that the drawings are the outcome of a highly detailed 3D model. Once submitted, the drawings typically need at least 2-3 weeks of review and comments from the classification society. At the same time, this information is also sent to the shipyard, enabling the detailed design process to begin and allowing procurement of plates, stiffeners and other equipment to proceed.

This naturally raised the question of whether we could shorten the approval time by sharing information with the classification society more quickly, thereby reducing the waiting period for drawing approval. In addition, the process of extracting and preparing DWG drawings is time-consuming and demands the effort of highly experienced engineers.

3.3. Moving to a Model-Based Design Process

Based on our experience with drawing preparation, we began considering an alternative approach: instead of spending significant time elaborating the drawings, we could redirect that effort toward developing a more complete and more detailed 3D model, Fig.5.

After the contractual GA is signed and we have a starting point, we model the tank boundaries and all the bulkheads. At this stage, we can present the first iteration to the classification society as an OCX export - replacing the traditional QC1 package of drawings. Together with the classification society, we can then agree on the required level of detail and focus on the most critical areas for approval.

Next, we add scantlings, girder layouts, and incorporate any comments that we receive from the classification society. This allows us to focus on the areas they consider most critical. We then proceed by adding the remaining structures and end connections. During the final quality check, our attention is placed on ensuring that all outstanding comments are resolved and properly reflected in the 3D model.

Communication with the classification society is expected to improve significantly under this approach, since instead of sending 20 or more drawings, only a single OCX file needs to be ex-

changed. Once all comments are incorporated, the 3D model is ready to be delivered to the classification society and, at the same time, shared with the shipyards for detailed design and production.

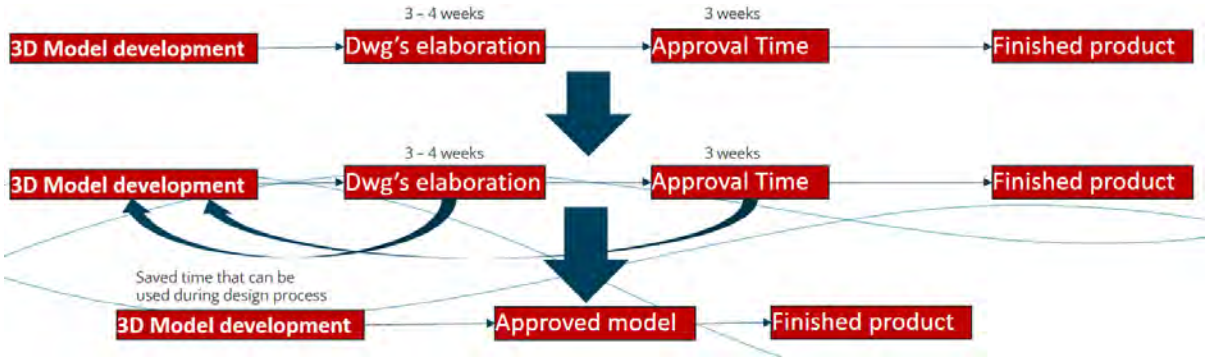


Fig.5: A 3D design process cuts time to delivery by shifting time spent on drafting to modelling and earlier involvement by the classification society

The focus moving forward will be to use the time currently spent on drawing development to create a more detailed and information-rich 3D model, which will serve as a foundation for the detailed design phase. Communication with the classification society should also become more efficient, as only a single OCX file needs to be exchanged between stakeholders.

4. Model-Based Approval

The classification approval process is an essential part of the ship design cycle. Traditionally, this process has relied on the exchange of 2D drawings between the shipyard and the classification society. Calculation models are either shared directly or constructed manually based on these 2D drawings. This approach often involves manual interpretation, frequent back-and-forth communication, and repeated documentation updates – all of which can be time-consuming and prone to inconsistencies.

With the introduction of 3D models and the OCX standard, now supported in tools like NAPA Designer, a model-based approval process has become possible. Instead of preparing extensive sets of 2D drawings, the 3D design model is submitted directly to the classification society using the OCX format, where it serves as the basis for structural review, rule calculations, and comments. This reduces the need to generate 2D drawings and eliminates the manual rebuilding of separate calculation models for approval purposes.

This section describes how model-based approval is applied in the collaboration between VARD, DNV, and NAPA. It highlights how feedback is managed directly within the 3D model and outlines the main benefits for the classification society.

4.1. Closing the comment feedback loop

Traditionally, DNV has issued two types of comments during the approval process: written comments included in an approval letter, and visual mark-ups placed on drawings. Written comments are typically used to provide general feedback, while mark-ups communicate specific requirements for rule compliance.

In the model-based approval process, both types of comments are now linked directly to the 3D model. When VARD submits the OCX model to DNV, the classification society reviews the model and provides comments using digital tools in a shared 3D environment. Comments are attached to specific locations or components in the model, making them more precise and easier to interpret, see Fig.6. for an example.

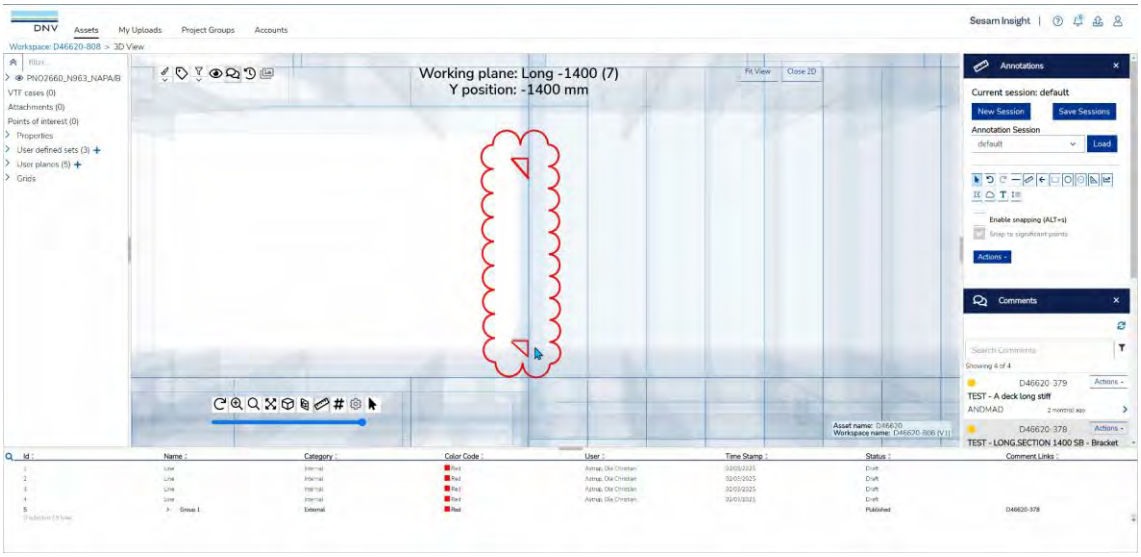


Fig.6: A visual mark-up added to the 3D model by the classification society

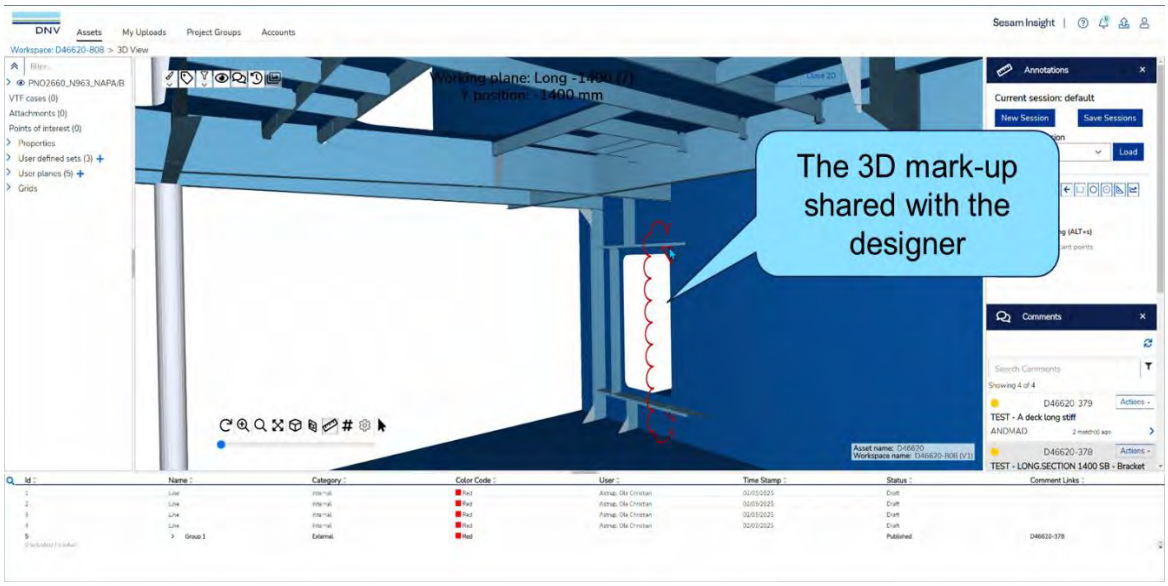


Fig.7: The 3D mark-up shared with the designer

This direct approach simplifies communication between the yard and the classification society. Designers can immediately identify where issues exist and implement changes directly in the 3D model. Once updates are made, a revised version of the model is submitted for review. Using comparison tools, DNV can quickly identify what has changed between revisions, allowing the approval engineer to focus on the relevant updates. This closes the feedback loop more efficiently and removes the need to issue revised 2D drawings for each modification.

4.2. Benefits for the Society

From the classification society's perspective, moving to 3D approval offers several clear advantages. Reviewing the full 3D model improves the overall understanding of the design, as spatial relationships and structural details are easier to interpret than in traditional 2D drawings.

The OCX model also enables rule-based checks to be performed directly on the 3D data, reducing manual effort and improving the quality of the review. For example, prescriptive rule calculations can be carried out using data extracted from the model, ensuring consistency and minimising the risk of missing information. In traditional workflows, it is still common to manually verify that scantlings in

calculation models match those shown in 2D drawings – a time-consuming and error-prone task. Model-based approval can eliminate this.

For more complex rule checks, such as those involving finite element analysis (FEA), the OCX model can also be reused to quickly build up the structural FEA model, even in external tools not natively supported by the classification society. This reuse leads to more accurate models and significantly reduces modelling time. As computing power and analysis tools continue to advance, rule checks are increasingly shifting into the 3D domain, making reliance on 2D drawings both inefficient and risky in terms of data loss.

In addition, model-based approval shortens the overall review cycle and improves traceability. All comments, mark-ups, and revisions are tracked within the model environment, making it easier to follow the history of decisions, feedback, and responses over the course of the project. This transparency benefits the classification society by reducing the need to re-check unchanged parts of the model, ensuring that reviews remain focused and efficient.

5. Conclusions

The transition from drawing-based to model-based approval represents a fundamental shift in ship design and classification processes. By positioning the 3D model as the single source of truth, VARD, DNV, and NAPA have demonstrated that information can be exchanged more efficiently, with greater accuracy and consistency than in traditional drawing-centric workflows. Using the OCX standard, classification approval can be carried out directly on the digital model, reducing the need for time-consuming drawing preparation, minimising the risk of discrepancies, and enabling clearer and more precise communication between designers and the classification society.

The benefits extend beyond approval. A continuously updated and information-rich 3D model creates a solid foundation for downstream processes such as detail design, procurement, and production. This reduces duplication of work, improves traceability of design changes, and shortens the overall design cycle. Ultimately, adopting the 3D model as the single source of truth supports a more efficient, transparent, and collaborative shipbuilding process.

6. Future work

This paper has demonstrated the benefits of model-based approval for both designers and classification societies during the approval stage of the design process. The next step is to extend this approach to the construction phase, exploring how the 3D model can support collaboration between the shipyard and the classification society's site survey team.

Using the 3D model as a single source of truth at the construction site has the potential to streamline inspections, improve transparency, and reduce the reliance on paper-based drawings and manual updates. Future work will therefore focus on how the OCX model can be integrated into yard production systems and survey workflows, ensuring that the same consistent and information-rich model supports design, approval, and construction activities.

References

ASTRUP, O.; AAE, O.; KUS, T.; UYANIK, O.; BITERLING, B.; BARS, T.; POLINI, M.; VIJAYA, G.; YU, K.; SEPPÄLÄ.T.; SON, M.J. (2022), *Moving towards model-based approval – the open class 3D exchange (OCX) standard*, Int. J. Maritime Eng. Part A

HALFHIDE, R. (2019), *APPROVED redraws requirements for class verification*, The Naval Architect

HUANG, Y.W.; TANG, Z.Y.; ZHANG, X.H.; ZHANG, H.; LIU, J.F. (2019), *Research on the Three-Dimensional Process Design Method of Shipbuilding Based on MBD Technology*, IOP Conf. Series: Materials Science and Engineering 616/1, https://iopscience.iop.org/article/10.1088/1757-899X/616/1/012029

MASUI, T.; SEPPÄLÄ, T.; SON, M.-J.; HUOTARI, J. (2023), *Collaborative Single Model Design Platform for Ships of Tomorrow*, COMPIT Conf., Drübeck

NAPA (2025), *3D-powered efficiency – how VARD delivered a ship design in just 10 weeks*, NAPA, https://www.napa.fi/case/vard/

SON, M.-J.; SEPPÄLÄ, T.; MERIKANTO, J.; AAE, O.; ASTRUP, O.C. (2022), *Utilization of OCX as Part of 3D Model Based Approval in Ship Design Process*, COMPIT Conf., Pontignano

Maritime AI for Trustworthy Decision-Making through Fundamental Research and Industry-Driven Applications

Svein Peder Berge, SINTEF Ocean, Trondheim/Norway, svein.berge@sintef.no
Pauline Røstum Bellingmo, SINTEF Ocean, Trondheim/Norway, pauline.bellingmo@sintef.no

Abstract

This paper presents six industry-driven AI applications that support trustworthy decision-making across the maritime value chain from ship design to operations. These use cases have been carefully selected based on Norway's strong domain expertise and demonstrate how AI can contribute to greener, safer, and more efficient maritime solutions. In addition, the paper discusses key research challenges related to maritime data and governance, relevant AI methods and models, the innovation process, and the adoption of AI within the maritime industry.

1. Introduction

The Norwegian Government has set ambitious targets for the maritime sector by 2030, a 50% reduction in shipping emission, *NMCE (2022)*, and a 50% increase in the export value of the maritime industry, *NMTIF (2020),* Maritime safety is also emphasized as a key component of national security. Over the past decade, artificial intelligence (AI) has significantly transformed society, industry, and everyday life. Advances in sensor technology, onboard internet access, and edge computing have expanded the potential for AI adoption across the maritime sector, *MacDonald and Martin (2024),* offering powerful tools to help achieve these national goals. While Norway is a relatively small player in AI, it is a global leader in maritime expertise. This unique position presents a strategic opportunity to develop AI solutions tailored to niche markets with global relevance. Success depends on identifying application areas where Norway's deep domain knowledge can be effectively combined with AI to deliver high-impact results. This paper presents six AI use cases, selected by Norwegian industry and research communities, that aim to reduce emissions, enhance safety, and improve operational efficiency.

The successful design and deployment of these AI applications require addressing key research challenges and socio-legal barriers. Trustworthy AI is essential for enabling safe and sustainable maritime operations. This includes AI systems that are transparent and explainable, robust and reliable, accountable, secure, fair, and respectful of privacy and intellectual property. In maritime contexts, poor AI model performance can compromise safety, reduce efficiency, and undermine trust. Therefore, explainability and transparency are critical for fostering human-AI collaboration, while accountability and security are vital in safety-critical operations. Moreover, the maritime industry faces several barriers to effective AI adoption, including limited access to high-quality data, weak data governance, regulatory and ethical concerns, cybersecurity risks, low user trust, and a shortage of AI expertise. Addressing both technical and non-technical challenges in parallel is essential. Overcoming these barriers will require close collaboration between industry stakeholders, policymakers, and the scientific community.

2. Trustworthy Maritime AI

Trustworthiness requires AI that is transparent and explainable, robust and reliable, accountable, safe and secure, responsible, fair and impartial, and respects privacy/IP, Fig.1. Using the framework from Deloitte, https://www.deloitte.com/uk/en/issues/generative-ai/trustworthy-ai.html, the trustworthy AI includes the following properties of the AI technology:

- Transparent and explainable. Users understand how technology is being leveraged, particularly in making decisions; these decisions are easy to understand, auditable, and open to inspection.
- Robust and reliable. The technology produces consistent and accurate outputs, withstands errors, and recovers quickly from unforeseen disruptions and misuse.

- Accountable. Policies are in place to determine who is responsible for the decisions made or derived with the use of AI technology.
- Safe and secure. The technology is protected from risks that may cause individual and/or collective physical, emotional, environmental, and/or digital harm.
- Responsible. The technology is created and operated in a socially responsible manner.
- Fair and impartial. The technology is designed and operated inclusively in an aim for equitable (fair) application, access, and outcomes.
- Private. User privacy is respected, and data is not used or stored beyond its intended and stated use and duration; users are able to opt-in/opt-out of sharing their data.

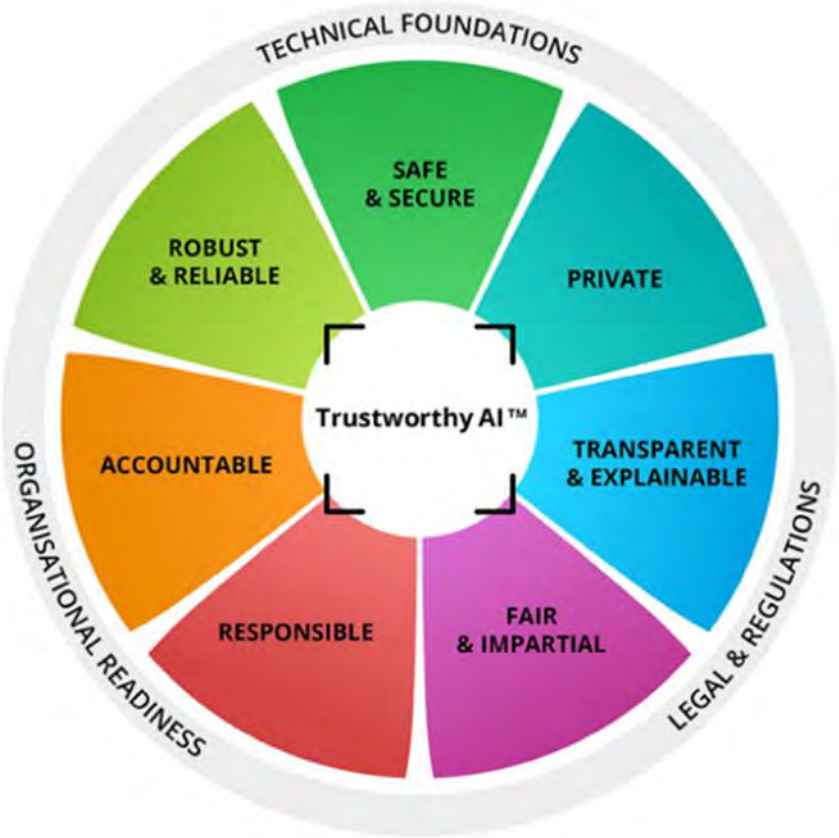


Fig.1: Deloitte – Trustworthy AI Framework

The adoption of AI in the maritime industry also involves several non-technological challenges, such as legal and regulatory frameworks, as well as organizational readiness. These factors are critical for the successful development, implementation, and integration of AI into daily operations.

3. Six Industry-Driven AI Applications in the Maritime Sector

Today, application of AI within the maritime sector is being researched and has already been applied within some applications. For instance, within autonomous ships, AI supports navigation and collision avoidance through situational awareness systems that process data from cameras and radars to detect and classify surrounding objects. Machine learning models trained on historical navigation and image data help assess collision risks and provide course-change recommendations to the crew, https://www.orca-ai.io/seapod/, https://www.kongsberg.com/discovery/navigation-positioning/seaaware/seaaware-ar-100/.

In voyage optimization, AI predicts energy consumption under varying conditions. By integrating weather forecasts and just-in-time arrival strategies, AI systems can recommend safer and more energy-efficient routes, *Jørgensen et al. (2022), GASS (2025)*.

Maritime security and surveillance also benefit from AI, which analyzes historical AIS data to detect abnormal vessel behavior and plan optimal voyage time slots, thereby reducing collision risks and enhancing situational awareness, https://vessel-ai.eu/. Operational efficiency is another area where AI is making a tangible impact. NYK Line, for example, employs condition-based maintenance for rotating machinery, leading to significant cost savings. For example, optimized lubrication services saved $1 million over 10 years, while repair and docking costs were reduced by $4 million. For vessels in service for 20 years, repair-related OPEX dropped from $10 million to $5 million, *MacDonald and Martin (2024)*.

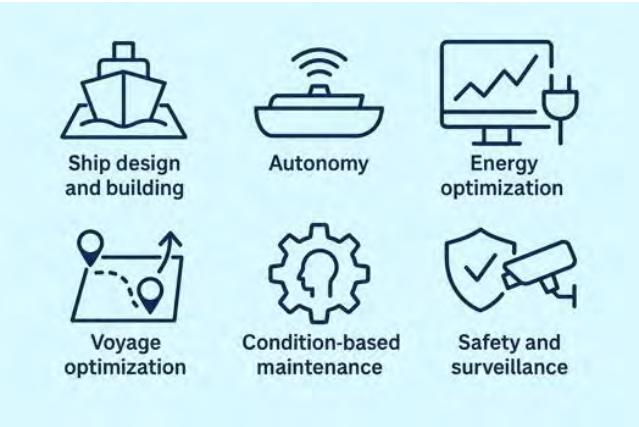


Fig.2: Maritime AI applications. Image from Copilot

While these advancements demonstrate the potential of AI, fully autonomous commercial shipping, where a vessel operates unmanned from port to port, remains out of reach. Human operators, whether onboard or in remote control centers, are still essential for oversight and final decision-making. To realize the vision of green, safe, and efficient maritime operations, further research and development is crucial.

Six key application areas for maritime AI have been identified based on industry needs, Fig.2. The overarching aim is to develop technologies and products that not only enhance ship design and operational efficiency but also deliver positive social and economic outcomes. These applications are expected to drive innovation and digitalization within the maritime sector, contributing to the creation of new jobs, increased export value, and the global competitiveness of Norwegian maritime technology. These six applications are described in the following sections.

3.1. AI Application #1 - Ship Design and Shipbuilding

The ship design process is inherently complex and relies on a wide array of specialized tools that often lack integration. This leads to manual workflows and repetitive tasks that are both time-consuming and error-prone. Customization for individual vessels and clients further increases complexity, especially during early-stage collaboration with multiple stakeholders. Current design methods typically assume ideal operating conditions, which do not reflect the realities of maritime operations. As a result, ships may be sub-optimally designed for real-world performance. Although high-fidelity simulation methods offer greater accuracy, their computational demands and setup complexity prevent them from being fully utilized in practice.

Needs: 1) Fast ship design tools for early evaluation, 2) Ship designs representing real operational conditions, 3) Improved design process efficiency, and 4) Innovative design solutions.

3.2. AI Application #2 - Autonomy

Ensuring the safety and efficiency of autonomous ships requires continuous monitoring and control from a Remote Operation Center. Operators rely on sensor data to maintain situational awareness

around each vessel. The complexity escalates when a single operator is responsible for multiple ships simultaneously, leading to information overload and a high cognitive load.

Needs: Advanced decision-support systems that intelligently aggregate and present sensor data in a clear and concise manner. Such systems should reduce cognitive load and enable operators to effectively monitor and manage several autonomous vessels at once.

3.3. AI Application #3 – Energy Management System

Integrating wind-assisted propulsion systems (WAPS) into ship operations introduces significant complexity. Selecting optimal set-points for sails, engines, energy storage, rudders, speed, and route must account for both vessel-specific characteristics and dynamic environmental conditions, *EMSA (2023)*. Accurately predicting the performance of these components under varying conditions remains a major challenge. Limited operational data for key WAPS components hinders both data-driven modelling and validation of physical simulations. Additionally, the aerodynamic forces from sails cause the vessel to operate with non-zero drift, rudder, and heel angles, even in steady-state conditions. This often pushes propulsion systems outside their optimal design range, making it difficult to determine the most efficient and safe operational strategy.

Needs: 1) High-accuracy prediction models for ship performance and energy systems under diverse operational conditions. 2) Optimization tools for energy management systems that support safe and fuel-efficient selection of set-points for WAPS-equipped vessels.

3.4. AI Application #4 – Voyage Optimization

Achieving true energy efficiency in voyage optimization depends on accurate predictions of ship performance under varying conditions. A key challenge lies in estimating ship resistance, particularly in rough weather, which is hydrodynamically complex and difficult to model, *Mittendorf et al. (2022)*. Additionally, ship performance is highly specific to individual vessel designs and operational profiles, making generalization across a fleet difficult. Uncertainty in weather forecasts further complicates optimization efforts, as even small deviations can significantly impact route planning and fuel consumption.

Needs: 1) Development of accurate and reliable ship performance models applicable across diverse vessel types. 2) Integration of weather uncertainty into voyage optimization algorithms to ensure robust and energy-efficient routing decisions.

3.5. AI Application #5 – Condition Monitoring and Predictive Maintenance

Machinery failure remains one of the most frequent and costly incidents in the shipping industry, making condition monitoring essential for improving system reliability and safety. However, current industrial practices often emphasize performance monitoring rather than predictive strategies, *Alfarizi et al. (2023)*. A major challenge in implementing predictive maintenance is the lack of standardized data formats and interoperability across maritime stakeholders. This is especially problematic for prognostic data, as failure records are rare and often insufficient for training robust AI models. Without consistent and high-quality data, predictive systems struggle to deliver accurate and actionable insights.

Needs: 1) Development of AI models capable of early fault detection and degradation prediction to enable just-in-time inspection and maintenance. 2) Incorporation of explicit uncertainty modelling to account for both the stochastic nature of degradation processes and limitations in data availability and model precision.

3.6. AI Application #6 – Security and Surveillance

Ensuring the safety and security of Norwegian waters involves both civil and military oversight—led by the Norwegian Coastal Administration through Vessel Traffic Services, and the Norwegian Joint Headquarters. These actors rely on decision support systems to monitor maritime operations and detect abnormal ship behaviour, a capability also critical for remote operation centres managing autonomous vessels. Effective surveillance requires the fusion of multiple data sources, including Automatic Identification System (AIS), radar, satellite imagery, and weather data. Moreover, building accurate and trustworthy anomaly detection systems is challenging due to the lack of annotated and validated training data for abnormal ship behaviour. This limits the reliability of AI models and hinders human-machine collaboration in safety-critical contexts.

Needs: Enhanced situational awareness and decision support for conventional, naval, and autonomous vessels by 1) Multi-source sensor fusion using shore-based and satellite sensors and 2) Improved accuracy and interpretability of anomaly detection models to ensure safety and trust.

4. Fundamental Maritime AI Research Challenges

Artificial Intelligence (AI) is built on the idea that machines can emulate human capabilities such as speaking, reasoning, learning, planning, and understanding. Traditionally, software systems relied on predefined algorithms based on theoretical and empirical physics to transform inputs into outputs. With the rise of AI and machine learning (ML), this paradigm has shifted. AI/ML uses data to infer the underlying relationships between inputs and outputs, effectively learning the model itself. This data-driven modelling allows systems to make predictions even without explicit knowledge of the underlying physics. However, the reliability of these "black-box" models depends heavily on the 4Vs of big data: Volume (amount of data), Velocity (speed of data generation), Variety (diversity of data types and Veracity (accuracy and trustworthiness of data). In the maritime industry, operational data from ships is often poorly structured and concentrated around a limited set of operating conditions. This clustering reduces the generalizability of AI models, meaning they may perform poorly outside the conditions they were trained on, potentially leading to incorrect or unsafe decisions.

The transformation from raw maritime data to reliable decision support in industrial applications involves a multi-step process, Fig.3. Each layer of this data-to-decision pyramid requires dedicated research, data acquisition, data structuring, post-processing, AI model development, and integration into end-user systems.

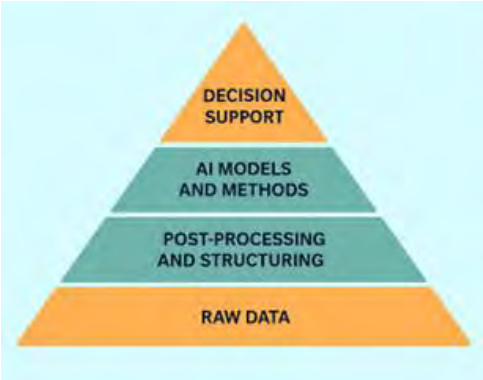


Fig.3: Transforming raw data to trustworthy decision support (Copilot)

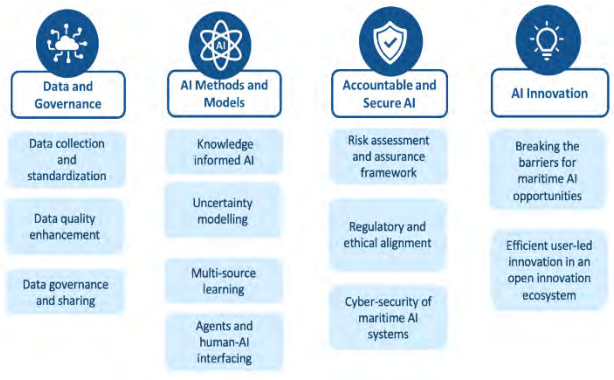


Fig.4: Fundamental maritime AI research challenges

At the base, raw data is collected from ship systems, designers, equipment providers, and industry stakeholders. This data is often proprietary and not openly shared due to competitive concerns. The next step involves post-processing and structuring the data to ensure it meets the quality and format requirements for AI modelling. This is essential for enabling robust and trustworthy AI applications. In

the third layer, AI models and methods are developed using the processed data. These tools should be open-source and adhere to the FAIR principles: Findable, Accessible, Interoperable, and Reusable, allowing the research community to build upon them (marked in green). Finally, the AI models are integrated into industrial products and services, where companies can protect their innovations through intellectual property rights. However, this full pipeline, from raw data to commercial deployment, is still immature and requires coordinated research across all stakeholders in the data value chain.

The fundamental maritime AI research challenges are categorized into four groups illustrated in Fig.4 and are described in the following subsections.

4.1. Data and Governance

The effective deployment of AI in maritime applications relies on access to high-quality, standardized, and securely shared data. Currently, maritime datasets are often fragmented, lack interoperability, and contain inconsistencies, which complicates both AI model training and real-time analytics, *Burlik et al. (2024)*. To address these challenges, systematic data collection and standardization efforts are essential. This includes the development of harmonized ontologies, knowledge graphs, and automated data ingestion pipelines that ensure semantic consistency and interoperability across AI-driven maritime systems. As demonstrated by *Li et al. (2024)*, knowledge graphs can significantly enhance automated decision-making by efficiently linking data from diverse maritime sources. However, many maritime stakeholders remain hesitant to share operational data due to competitive pressures, legal constraints, and concerns over data ownership and misuse. This reluctance poses a major barrier to collaborative innovation in AI, *Zhao et al. (2024)*. To overcome this, a secure and standardized data governance framework is needed - one that promotes trust, facilitates interoperability, and supports scalable AI adoption throughout the maritime sector.

To enable the deployment of high-quality, secure, and ethically compliant data in maritime applications, there is need for research on:

- Systematic data collection and standardization,
- Data quality enhancement, and
- Data governance and sharing.

4.1.1. Data Collection and Standardization

This area focuses on systematically collecting, mapping, and structuring maritime datasets from a wide range of sources to support AI-driven applications in ship design, autonomy, energy management, voyage optimization, condition monitoring, and security. Relevant data types include: ship performance data (e.g., engine metrics, fuel consumption, speed, alarms), navigational data (e.g., camera feeds, radar, satellite imagery, AIS), metocean data, simulation and model test data, and ship design data (e.g., specifications, drawings, 2D/3D models).

4.1.2. Data Quality Enhancement

Improving the quality of maritime datasets is essential for reliable AI performance and can be achieved by e.g.:

- Developing methods to quantify uncertainty and assess dataset completeness (e.g., detecting sensor drift, missing data, or outliers).
- Applying techniques such as interpolation and sensor fusion to repair and enhance low-quality or incomplete datasets.
- Leveraging physics-based simulations and experimental model test data to supplement real-world datasets, especially in scenarios with limited data access (e.g., early design phases or extreme conditions).

- Enriching training datasets with domain knowledge, and optimizing sample design by analysing data coverage, tail behaviour, and representativeness.

4.2.3. Data Governance and Sharing

Fostering collaboration and ensure secure, standardized, and legally compliant data sharing across maritime stakeholders, can be achieved by the following measures:

- Implementing a data governance framework based on the International Data Space Association reference architecture.
- Developing trusted data-sharing mechanisms that maintain stakeholder control over sensitive data while ensuring interoperability, *Kuster (2021)*.
- Applying federated learning techniques in cases where direct data sharing is restricted, enabling collaborative model training while preserving data privacy and protecting proprietary information.

4.3. AI Models and Methods

Traditionally, the maritime industry has relied on knowledge-based systems to support decision-making in ship design and operations. While physical models offer robustness and explainability, they often fall short in capturing the complexity of real-world operational conditions. AI presents a promising alternative by learning patterns directly from data, and in some cases, outperforming simplified physical models. However, AI models face challenges related to generalization, transparency, and extrapolation to unfamiliar scenarios - issues that can lead to inaccurate predictions and reduced trust in decision support systems. Additionally, the limited availability and quality of operational data in maritime contexts make it difficult to train reliable AI models using operational data alone. To overcome these limitations, extensive datasets from laboratory experiments and simulations can be used to complement operational data. This hybrid approach enables the development of more accurate and robust models, even in scenarios where real-world data is scarce or unavailable. Ultimately, these challenges highlight the need for AI models that integrate both theoretical and empirical knowledge, paving the way for more trustworthy and effective decision support in maritime applications.

To develop trustworthy AI solutions for maritime applications, we highlight four key research areas:

- Knowledge-informed AI
- Uncertainty modelling
- Multi-source learning
- Agents and human-AI interfacing

4.3.1. Knowledge-Informed AI

The objective is to develop accurate, robust, and explainable data-driven models for maritime applications by integrating domain-specific maritime knowledge with AI, using hybrid and causality-aware approaches. The type of domain knowledge varies across application areas. For example, in ship design, it may include requirements related to hydrodynamics, hydrostatics, aerodynamics, structural integrity, and operational profiles. This knowledge can be incorporated into AI models as informative priors, hard constraints, or causal structures. Hard-constrained models are developed to respect fundamental physical laws, building on recent advances in structure-preserving modelling, *Greydanus et al. (2019), Cranmer et al. (2020), Eidnes et al. (2023).* Knowledge-informed loss functions penalize predictions that contradict empirical knowledge and help learn coefficients in established models, *Raissi et al. (2019), Cai et al. (2021)*. Causal reasoning will be used to improve generalization, provide actionable explanations, *Beckers (2022)*, enhance transportability, and enable principled fusion of data from experimental and observational sources, *Pearl and Bareinboim (2022)*.

4.3.2. Uncertainty Modelling

Accurately capturing and propagating uncertainty is essential for building trustworthy AI models in maritime applications. Data uncertainty can be reflected in model outputs using techniques such as Monte Carlo sampling and probabilistic model layers, *Loquercio et al. (2020)*. Uncertainty modelling can be achieved by developing methods to assess and benchmark uncertainty estimates - particularly the calibration of prediction intervals and posterior distributions - across a wide range of existing approaches. These include principled techniques such as Bayesian neural networks, *Gal and Ghahramani (2016)* and Gaussian processes, *Rasmussen (2004),* as well as heuristic methods, *Wenzel et al. (2020)*. Furthermore, uncertainty can be modelled by incorporating prior domain knowledge into uncertainty estimation, tailored to specific maritime applications. By adapting these methods to hybrid models, this enables the propagation of both data and model uncertainties within a unified framework, *Psaros et al. (2023)*.

This ensures that different types of uncertainty - whether stemming from data quality, model assumptions, or operational variability - are consistently accounted for in AI-driven decision support systems.

4.3.3. Multi-Source Learning

To improve the accuracy of AI models in maritime applications—particularly where operational data is scarce - this research area focuses on leveraging multiple data sources, including simulation data, model test data, and real-world operational data. This approach is especially relevant for emerging ship types or vessels equipped with energy-saving technologies, where operational data may be limited or unavailable. Example of multi-source learning methods include:

- Multi-fidelity learning which enable models to learn from data of varying quality and resolution, *Meng et al. (2021a,b)*.
- Data-driven model order reduction techniques can be applied to develop efficient models suitable for real-time predictions, *Conti et al. (2024)*.
- Transfer learning can be integrated with the multi-fidelity framework, allowing knowledge gained from well-documented vessels to be applied to ships with limited data, *Song and Tartakovsky (2022)*.

4.3.4. Agents and Human-AI Interfacing

Recent advancements in reasoning models, *DeepSeek-AI (2025)*, https://openai.com/nb-NO/, demonstrate significant potential for AI to assist humans in complex tasks—ranging from information selection and structuring to decision support. While these technologies are still in early stages and lack formal safety guarantees, their capabilities open up promising opportunities in maritime domains such as ship design and autonomous operations.

This research area explores how reasoning models can be effectively combined with established tools - including analytical models, simulators, clustering algorithms, and optimization techniques - to create intuitive and reliable decision support systems for human operators. The goal is to develop AI agents that enhance human decision-making by providing relevant insights, guiding design choices, and adapting to user feedback. This area includes use of:

- Large language models (LLMs) as interfaces for physical simulation models.
- AI agents that propose ship design alternatives and link them to appropriate physical models.
- Methods that organize and prioritize information, learning continuously from human interaction and feedback.

By integrating these emerging AI capabilities with traditional engineering tools, we can build human-AI interfaces that are both user-friendly and capable of delivering meaningful support in complex maritime environments.

4.4. Accountable and Secure AI

The integration of AI into maritime systems introduces a range of novel risks. These include unsafe or suboptimal decisions caused by inaccurate predictions, poor human-AI collaboration, cybersecurity vulnerabilities, and unclear accountability—all of which can have serious consequences for the industry, the environment, and public safety, *Bengio et al. (2024)*. Current regulations, standards, organizational frameworks, and security systems are not yet equipped to fully address these risks, *Durlik et al. (2024)*. The development of compliance and assurance standards for maritime AI has been slow and fragmented. As AI technologies evolve, the maritime sector faces increasing challenges in aligning with both existing and emerging regulatory requirements. Although the industry has shown adaptability under regulatory uncertainty, there remains a lack of clarity regarding how, when, and where new assurance mechanisms should be applied. To ensure safe and reliable deployment of AI, there is a pressing need for comprehensive risk assessments that capture the full cause-and-effect chain - from data sources and AI models to the operational decisions made by AI systems. These assessments must be dynamically aligned with evolving assurance standards and regulatory frameworks. Full compliance requires sensitivity to different stages of AI systems (e.g., data governance, model deployment, liability mechanisms), diverse application contexts (e.g., ship design, autonomy, surveillance), and varied governance regimes (national/international, maritime/non-maritime, public/private law).

To ensure the safe, secure, ethical, and trustworthy deployment of AI in the maritime sector, three key research areas have been identified:

- Risk assessment and assurance framework
- Regulator and ethical alignment
- Cybersecurity challenges

4.4.1. Risk Assessment and Assurance Framework

This research area includes developing a comprehensive framework for assessing risks in AI-enabled maritime systems. The framework should leverage established safety standards such as ISO/IEC 22989 and ISO/IEC 23053, DNV standards and practices (e.g., DNV-RP-0671 Assurance of AI-Enabled Systems), and IMO guidelines for AI assurance. The framework should define key performance indicators (KPIs) and acceptance criteria—including anomaly detection accuracy and autonomy scenario coverage—to ensure AI components meet measurable thresholds for safety and reliability. More, the framework should develop a dedicated Maritime AI Assurance Toolkit, including process guidelines, templates, and checklists to support verification and validation of AI systems prior to deployment.

4.4.2. Regulatory and Ethical Alignment

There is need for mapping the complex and evolving regulatory landscape surrounding maritime AI. This includes legacy maritime regulations, emerging AI-specific standards (e.g., EU AI Act, GDPR), international and transnational law, liability and insurance frameworks, and ethical guidelines across jurisdictions. This mapping should identify regulatory gaps, overlaps, and critical issues affecting AI design and deployment. Based on this mapping, comprehensive guidelines can be developed to help stakeholders navigate key areas such as transparency, accountability, and human oversight. Policy recommendations will be shared with relevant authorities, including the Norwegian Maritime Authority and EU-level bodies, to support future standardization and ensure a coherent regulatory framework.

4.4.3. Cybersecurity Challenges

Despite significant progress in maritime digitalization, AI-driven systems remain highly vulnerable to cyber threats. These include GNSS spoofing, adversarial attacks on AIS, and data tampering, all of which pose serious risks to operational safety and system integrity, *Neumann (2024)*. Current cybersecurity frameworks in the maritime domain are not adequately equipped to handle these emerging threats. They lack:

- Real-time AI-enabled threat detection mechanisms
- Standardized security guidelines for AI in maritime monitoring
- Recommended architectures and best practices for security-by-design
- Robust mitigation strategies to ensure safe and resilient AI deployment

Moreover, the interdependencies between cyber, maritime, and space domains are not sufficiently addressed in existing regulatory structures. This fragmentation and lack of coordination across frameworks hinder the development of comprehensive and future-proof cybersecurity solutions for maritime AI systems. To address these challenges, there is need for a maritime AI cybersecurity blueprint comprising:

- Security-by-design principles to ensure robustness against cyberattacks and system failures
- Assessment and mitigation tools aligned with the EU AI Act, ISO/IEC 22989, and IMO guidelines on maritime cyber risk management
- Resilience techniques, including intrusion detection, anomaly detection, and dynamic threat response strategies

The blueprint can support the secure, resilient, and regulation-compliant adoption of AI across maritime stakeholders.

4.5. AI Innovation in Maritime Sector

The integration of AI as an enabling technology in maritime innovation raises fundamental epistemological questions, particularly at the intersection of open innovation and the exploitation of emerging AI capabilities. In publicly funded research ecosystems involving diverse maritime stakeholders, open innovation faces significant barriers to effective collaboration, *Ankrah and Al-Tabbaa (2015)*. AI adds further complexity by introducing challenges related to trust, knowledge gaps, skill disparities, and the technical and design characteristics of novel solutions, *Marocco et al. (2024)*. The cross-disciplinary and multi-stakeholder nature of maritime AI innovation makes it difficult to fully harness the transformative potential of these technologies. To address these challenges and foster impactful innovation, two main research activities have been defined:

- Breaking barriers to maritime AI opportunities
- User-led innovation in an open ecosystem

4.5.1. Breaking Barriers to Maritime AI Opportunities

This activity focuses on developing a comprehensive framework for innovation management, including:

- Innovation strategy and IP utilization plans
- Training programs and collaborative arenas to foster an innovation culture
- Methodologies for identifying and mitigating non-technical barriers in technical development
- Integration of insights from data, AI models and models, and validation

Special emphasis should be placed on establishing co-creation arenas, *Ertz (2024),* that promote shared value creation, continuous learning, and a unified understanding across disciplines. These arenas can also serve as platforms for documenting collaborative processes and outcomes.

4.5.2. User-Led Innovation in an Open Ecosystem

This initiative leverages the principles of action research, *Clark et al. (2020)*, a methodology well-suited for exploring innovation in dynamic, real-world environments. Action research enables iterative cycles of planning, implementation, analysis, and reflection, making it particularly effective for studying emerging technologies. In the context of maritime AI, this approach can be used to investigate the barriers, challenges, and opportunities associated with collaborative development. By integrating multi-stakeholder perspectives and real-time feedback, action research can generate actionable insights into organizational and managerial processes. This supports stakeholders in navigating the complex, unpredictable nature of AI-driven innovation and foster a more adaptive and inclusive innovation ecosystem.

5. Conclusion

The transformative potential of AI in the maritime industry depends on sustained, long-term research - both in AI technologies and innovation processes. To unlock this potential, it is essential to develop and deploy trustworthy AI that supports decision-making across the entire maritime value chain, from ship design to daily operations. Trustworthy AI must be:

- Transparent and explainable
- Robust and reliable
- Accountable, safe, and secure
- Responsible, fair, and impartial
- Respectful of privacy and intellectual property

Achieving these qualities requires iterative, cross-disciplinary collaboration among researchers, industry stakeholders, and regulators. Standardized frameworks for data quality, data governance, risk assessment, and assurance are critical to ensure that AI systems meet measurable thresholds for safety and reliability. Ultimately, trustworthy AI can serve as a powerful tool to assist crews onboard or operators in remote control centres in making better, more informed decisions. However, until AI systems reach a fully validated level of trust, humans must remain in the loop to ensure safe and responsible operations.

6. Future Outlook

The Norwegian Government has this spring announced a major investment in AI, committing 1.2 billion NOK (100 million EUR) to support six AI research centres over the next five years. These centres will collectively together cover the three major tracks:

- Societal consequences - research on societal consequences of AI
- Technology - research-based development of AI and AI-relevant technologies
- Innovation - research on how to use AI and any other digital technologies

The maritime actors SINTEF Ocean, Kongsberg and DNV are involved in the Norwegian Centre on AID for Decision (AID), one of the six centres, and will focus on relevant maritime use cases. This centre will also cover other sectors like energy, process, health and develop AI technology for multi-sectional use. In addition, there is a dedicated centre for Maritime AI, with a funding of 100 million NOK with focus on AI-relevant technology and AI innovation processes for maritime industry. The centre will contribute developing AI technology supporting greener, safer and more efficient maritime operations. The centre has partners from the whole maritime value chain including ship designer, shipyard, equipment providers, ship owners and operators, class society, universities and research institutes.

In a broader context beyond ship design and operation, AI is increasingly being applied in port environments to automate key processes such as berth allocation, cargo tracking, and traffic management. AI-powered digital twins enable simulation of port operations and infrastructure changes, helping to identify and prevent bottlenecks while optimizing logistics. Edge computing and maritime 5G are critical enabling technologies, providing low-latency control for cranes and unmanned surface vessels (USVs). These technologies also support remote operations and significantly enhance cybersecurity across maritime systems.

Acknowledgements

This study is funded by the Research Council of Norway through the projects Fusing AI and Ship Hydrodynamics for Next-Gen Voyage Optimization (FUSE) with project number 352781 and SFI AutoShip: Autonomous Ships for Safe and Sustainable Operations with project number 309230.

References

ALFARIZI, M.G.; TAJIANI, B.; VATN, J.; YIN, S. (2023), *Optimized Random Forest Model for Remaining Useful Life Prediction of Experimental Bearings*, IEEE Trans. Industrial Informatics 19(6), pp.7771–7779

ANKRAH, S.; AL-TABBAA, O. (2015), *Universities–industry collaboration: A systematic review*, Scandinavian J. Management 31(3), pp.387–408

BECKERS, S. (2022), *Causal Explanations and XAI*, 1st Conf. Causal Learning and Reasoning, pp.90–109, https://proceedings.mlr.press/v177/beckers22a.html

BENGIO, Y.; HINTON, G.; YAO, A.; SONG, D.; ABBEEL, P.; DARRELL, T.; HARARI, Y.N.; ZHANG, Y.Q.; XUE, L.; SHALEV-SHWARTZ, S.; HADFIELD, G.; CLUNE, J.; MAHARAJ, T.; HUTTER, F.; BAYDIN, A.G.; MCILRAITH, S.; GAO, Q.; ACHARYA, A.; KRUEGER, D.; DRAGAN, A.; TORR, P.; RUSSELL, S.; KAHNEMAN, D.; BRAUNER, J.; MINDERMANN, S. (2024) *Managing extreme AI risks amid rapid progress*, Science 384(6698), pp.842–845

CAI, S.; MAO, Z.; WANG, Z.; YIN, M.; KARNIADAKIS, G. (2021), *Physics-informed neural networks (PINNs) for fluid mechanics: a review*, Acta Mechanica Sinica 37(12), pp.1727–1738

CLARK, J.S.; PORATH, S.; THIELE, J.; JOBE, M. (2020), *Action Research*, Kansas State University Libraries: New Prairie Press, https://newprairiepress.org/ebooks/34/

CONTI, P.; GUO, M.; MANZONI, A.; FRANGI, A.; BRUNTON, S.L.; KUTZ, J.N. (2024), *Multi-fidelity reduced-order surrogate modelling*, Proc. Royal Society A: Mathematical, Physical and Engineering Sciences 480(2283), p. 20230655

CRANMER, M.; GREYDANUS, S.; HOYER, S.; BATTAGLIA, P.; SPERGEL, D.; HO, S. (2020), *Lagrangian Neural Networks*, arXiv, https://doi.org/10.48550/arXiv.2003.04630

DEEPSEEK-AI (2025), *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning*, arXiv, https://doi.org/10.48550/arXiv.2501.12948

DURLIK, I.; MILLER, T.; KOSTECKA, E.; TUNSKI, T. (2024), *Artificial Intelligence in Maritime Transportation: A Comprehensive Review of Safety and Risk Management Applications*, Applied Sciences 14(18), p. 8420

EIDNES, S.; STASIK, A.J.; STERUD, C.; BOHN, E.; RIEMER-SORENSEN, S. (2023), *Pseudo-Hamiltonian neural networks with state-dependent external forces*, Physica D: Nonlinear Phenomena, 446, p.133673, https://doi.org/10.1016/j.physd.2023.133673

EMSA (2023), *Potential of wind-assisted propulsion for shipping*, European Maritime Safety Agency, https://emsa.europa.eu/publications/item/5078-potential-of-wind-assisted-propulsion-for-shipping.html

ERTZ, M. (2024), *Co-Creation*, Encyclopedia 4(1), pp.137–147, https://doi.org/10.3390/encyclopedia4010012

GAL, Y.; GHAHRAMANI, Z. (2016), *Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning*, 33rd Int. Conf. Machine Learning, https://proceedings.mlr.press/v48/gal16.html

GASS (2025), *Green AI for Sustainable Shipping, to reduce GHG emissions*, GASS-project, https://www.gass-project.com

GREYDANUS, S.; DZAMBA, M.; YOSINSKI, J. (2019), *Hamiltonian Neural Networks', in Advances in Neural Information Processing Systems*, Curran Associates Inc., https://proceedings.neurips.cc/paper/2019/hash/26cd8ecadce0d4efd6cc8a8725cbd1f8-Abstract.html

JØRGENSEN, U.; BELINGMO, P.R.; MURRAY, B.; BERGE, S.P.; POBITZER, A. (2022), *Ship route optimization using hybrid physics-guided machine learning*, J. Physics: Conference Series, 2311(1), p.012037, https://doi.org/10.1088/1742-6596/2311/1/012037

KUSTER, A. (2021), *Maritime Data Spaces Provides Trusted Environment for Industry Data Sharing*, International Data Spaces, https://internationaldataspaces.org/maritime-data-spaces-provides-trusted-environment-for-industry-data-sharing/

LI, Y.; LIU, X.; WANG, Z.; MEI, Q.; XIE, W.; YANG, Y.; PENG, W. (2024), *Construction of a large-scale maritime element semantic schema based on knowledge graph models for unmanned automated decision-making*, Frontiers in Marine Science 11, https://doi.org/10.3389/fmars.2024.1390931

LOQUERCIO, A.; SEGU, M.; SCARAMUZZA, D. (2020), *A General Framework for Uncertainty Estimation in Deep Learning*, IEEE Robotics and Automation Letters 5(2), pp.3153–3160

MacDONALD, F.; MARTIN, D. (2024), *Opportunities and Obstacles in the Maritime AI Boom*, https://thetius.com/free-report-beyond-the-horizon-opportunities-and-obstacles-in-the-maritime-ai-boom/

MAROCCO, S.; BARBIERI, B.; TALAMO, A. (2024), *Exploring Facilitators and Barriers to Managers' Adoption of AI-Based Systems in Decision Making: A Systematic Review*, AI 5(4), pp.2538–2567, https://doi.org/10.3390/ai5040123

MENG, X.; WANG, Z.; FAN, D.; TRIANTAFYLLOU, M.S.; KARNIADAKIS, G.E. (2021), *A fast multi-fidelity method with uncertainty quantification for complex data correlations: Application to vortex-induced vibrations of marine risers*, Computer Methods in Applied Mechanics and Engineering 386, https://doi.org/10.1016/j.cma.2021.114212

MENG, X.; BABAEE, H.; KARNIADAKIS, G.E. (2021), *Multi-fidelity Bayesian neural networks: Algorithms and applications*, J. Computational Physics 438

MITTENDORF, M.; NIELSEN, U.D.; BINGHAM, H.B. (2022), *Data-driven prediction of added-wave resistance on ships in oblique waves - A comparison between tree-based ensemble methods and artificial neural networks*, Applied Ocean Research 118, https://doi.org/10.1016/j.apor.2021.102964

NEUMANN, T. (2024), *Cybersecurity in Maritime Industry*, TransNav, Int. J. Marine Navigation and Safety of Sea Transportation 18(4), pp.765–774, https://doi.org/10.12716/1001.18.04.02

NMCE (2022), *Norway's Climate Action Plan for 2021–2030*, Norwegian Ministry of Climate and Environment, https://www.regjeringen.no/contentassets/a78ecf5ad2344fa5ae4a394412ef8975/en-gb/pdfs/stm202020210013000engpdfs.pdf

NMTIF (2020), *The Norwegian Government's action plan for export*, Action Plan W-0035 E, Norwegian Ministry of Trade, Industry and Fisheries, https://www.regjeringen.no/contentassets/a60f7f916d424bb5b79903ebcdba1ace/208414-nfd-handlingsplan-engelsk-web.pdf

PEARL, J.; BAREINBOIM, E. (2022), *External Validity: From Do-Calculus to Transportability Across Populations*, Probabilistic and Causal Inference: The Works of Judea Pearl, Association for Computing Machinery, pp.451–482

PSAROS, A.F.; MENG, X.; ZOU, Z.; GUO, L.; KARNIADAKIS, G.E. (2023), *Uncertainty quantification in scientific machine learning: Methods, met-rics, and comparisons*, J. Comp. Physics 477

RAISSI, M.; PERDIKARIS, P.; KARNIADAKIS, G.E. (2019), *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*, J. Computational Physics 378, pp.686–707

RASMUSSEN, C.E. (2004), *Gaussian Processes in Machine Learning*, https://www.researchgate.net/publication/41781206_Gaussian_Processes_in_Machine_Learning

SONG, D.H.; TARTAKOVSKY, D.M. (2022), *Transfer Learning on Multifidelity Data*, J. Machine Learning for Modeling and Computing 3(1), pp.31–47

WENZEL, F.; ROTH, K.; VEELING, B.S.; ŚWIĄTKOWSKI, J.; TRAN, L.; MANDT, S.; SNOEK, J.; SALIMANS, T.; JENATTON, R.; NOWOZIN, S. (2020), *How Good is the Bayes Posterior in Deep Neural Networks Really?*, arXiv, https://doi.org/10.48550/arXiv.2002.02405

ZHAO, G.; XIE, X.; WANG, Y. ; LIU, S.; JONES, P.; LOPEZ, C. (2024), *Barrier analysis to improve big data analytics capability of the maritime industry: A mixed-method approach*, Technological Forecasting and Social Change 203, https://doi.org/10.1016/j.techfore.2024.123345

Seakeeping and Structural Simulation Using Open Simulation Platform with Live Data Integration

Ulrik Jørgensen, SINTEF Ocean, Trondheim/Norway, ulrik.jorgensen@sintef.no
Svein Peder Berge, SINTEF Ocean, Trondheim/Norway, svein.berge@sintef.no
Kristoffer Eide, SINTEF Ocean, Trondheim/Norway, kristoffer.eide@sintef.no
Stian Skjong, SINTEF Nordvest, Ålesund/Norway, stian.skjong@sintef.no

Abstract

This paper presents a digital twin of the research vessel R/V Gunnerus using the Open Simulation Platform (OSP) to predict seakeeping capabilities and structural responses during operation. A model is developed, integrating live vessel- and environmental data to give decision-making support with regards to motions and structural loads in real-time. To enable two-way data exchange, an MQTT-based communication solution has been developed, allowing the vessel model to both receive and provide data to connected systems. The approach demonstrates the potential of OSP for enhancing operational awareness and decision-making support in marine operations, while showcasing the benefits of modular, standards-based simulation architectures in maritime applications.

1. Introduction

Digital twins are rapidly gaining traction in the maritime industry as tools for real-time decision support, performance prediction, and safety assurance. By virtually replicating the physical behaviour of vessels and systems, digital twins enable continuous monitoring and analysis under operational conditions. The Open Simulation Platform (OSP), https://opensimulationplatform.com/, is a collaborative initiative aimed at supporting this vision by providing a standards-based framework for co-simulation of maritime systems using the Functional Mock-up Interface, https://fmi-standard.org/. It facilitates modular, reusable, and interoperable simulation components that can be integrated into real-time systems.

This paper presents a proof-of-concept implementation of a digital twin for the research vessel R/V Gunnerus using OSP. The objective is to demonstrate how existing OSP reference models can be extended and coupled with live data to enable real-time seakeeping and structural response simulations. The digital twin ingests real-world measurements from onboard sensors, including GNSS, wave radar, wind sensors, and propulsion systems, and uses these to drive hydrodynamic and structural models of the vessel. In return, the simulation outputs provide enhanced situational awareness and decision-making support during marine operations.

To enable this live data exchange, we developed a dedicated MQTT-based network FMU, enabling two-way communication between the physical vessel and the simulation system. This architecture allows the digital twin to operate as part of an integrated onboard environment, exchanging data in real time with other systems through a lightweight, scalable, and standards-compliant solution.

The work presented is based on research within the EDINAF project, https://edinaf.eu/, and recent findings on coordinate transformation techniques in maritime co-simulations, *Sadjina et al. (2024)*. Together, these developments contribute to a robust framework for deploying operational digital twins onboard ships using open standards and publicly available model libraries. The main contributions of this paper are threefold: (1) the demonstration of an onboard digital twin for seakeeping and structural response simulation using OSP and live sensor data; (2) the development and integration of a new MQTT FMU that enables real-time bidirectional communication between physical systems and simulation components; and (3) the extension of OSP reference models to include structural response estimation, together with coordinate transformation FMUs that enable seamless geodetic-to-local frame mapping. These developments enhance the practical applicability of OSP in real-world maritime operations and support the broader adoption of modular digital twin architectures.

1.1 Methodology

This work is based on a proof-of-concept implementation of a digital twin for R/V Gunnerus, developed using the Open Simulation Platform. The methodology involves the integration of real-time data streams with a simulation model to evaluate vessel motions and structural responses. As part of the study, we assessed and compared available solutions for simulation execution and data communication, ultimately implementing an MQTT-based approach to enable two-way communication between the live vessel and the digital twin. The selected architecture was tested under real conditions to demonstrate feasibility and evaluate performance within a modular and standards-compliant approach.

2. Modelling

In this work, we primarily utilize models available from the Open Simulation Platform (OSP) reference model library, https://open-simulation-platform.github.io/demo-cases, to ensure consistency with established standards and to leverage existing validated components. Where these reference models do not adequately address the specific requirements of our case, relevant theoretical background is presented to provide the necessary foundation. Based on this theory, we develop extended versions of existing Functional Mock-up Units (FMUs) or implement new FMUs tailored to our simulation needs. This approach ensures both alignment with the OSP ecosystem and the flexibility to address domain-specific challenges.

2.1. Seakeeping

To accurately predict vessel behaviour in various sea states, a fundamental understanding of seakeeping theory is essential. This section outlines the theoretical basis for modelling ship motions in waves, including the hydrodynamic principles and response amplitude operators (RAOs) used in the simulation of R/V Gunnerus, https://www.ntnu.edu/oceans/gunnerus. To this end we consider a maritime vessel as seen in Fig.1.

		BODY		NED
DOF		Forces and moments	Linear and angular velocities	Positions and Euler angles
1	Motions in the x_b-direction (surge)	X	u	x^n
2	Motions in the y_b-direction (sway)	Y	v	y^n
3	Motions in the z_b-direction (heave)	Z	w	z^n
4	Rotation about the x_b-axis (roll)	K	p	ϕ
5	Rotation about the y_b-axis (pitch)	M	q	θ
6	Rotation about the z_b-axis (yaw)	N	r	ψ

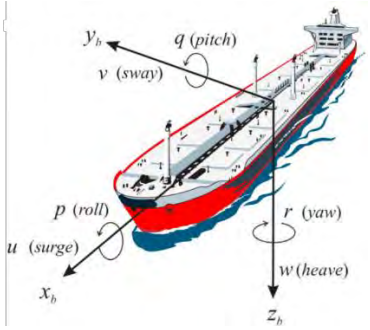


Fig.1: Definition of marine vessel state variables. Courtesy: *Fossen (2021)*

Now consider the corresponding 6 degrees of freedom (6-DOF) vectorial equation of motions for a marine vessel as presented in *Fossen (2021)*:

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= \boldsymbol{J}_{\boldsymbol{\Theta}}(\boldsymbol{\eta})\boldsymbol{\nu} \\ \boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) &= \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{wind}} + \boldsymbol{\tau}_{\text{waves}}. \end{aligned} \tag{1}$$

$\boldsymbol{\eta}$ is the North-East-Down (NED) position and attitude vector, $\boldsymbol{\nu}$ the corresponding linear- and angular velocity vector, $\boldsymbol{J}_{\boldsymbol{\Theta}}$ a matrix that consists of an Euler angle rotation matrix and a translational matrix on the diagonal, $\boldsymbol{M}$ the vessel mass matrix, $\boldsymbol{C}$ the Coriolis matrix, $\boldsymbol{D}$ the damping matrix, $\boldsymbol{g}$ the gravity vector and $\boldsymbol{\tau}, \boldsymbol{\tau}_{\text{wind}}, \boldsymbol{\tau}_{\text{waves}}$ general, wind and wave forces acting one the vessel, respectively. Note that the general force vector $\boldsymbol{\tau}$ is usually utilized to manipulate the vessel position and velocities for instance through the use of propellers and rudders. The seakeeping task consists in determining the position and attitude and corresponding linear and angular velocities given the input forces.

To solve the seakeeping task we turn to the Vessel Response (VERES) program, *Fathi (2018), Fathi and Hoff (2017),* which provides a robust framework for vessel seakeeping analysis, grounded in linear potential theory and strip theory simplifications. It assumes the vessel undergoes harmonic oscillations at the frequency of wave encounter, neglecting transient and hydro-elastic effects. The model presumes a linear relationship between wave amplitude and vessel response, which holds under moderate sea states but may deviate in severe conditions due to nonlinear phenomena such as slamming and deck wetness. VERES applies the principle of superposition to compute motions and loads across a sea state, treating the fluid as homogeneous, incompressible, irrotational, and non-viscous. The hull is assumed to be slender and symmetrical about the centreline, enabling the reduction of the three-dimensional hydrodynamic problem to a series of two-dimensional cross-sectional analyses. These properties fit nice with our problem; hence we utilize VERES to produce transfer functions suitable for solving the first order wave-induced forces for a wide range of sea states.

While VERES provides transfer functions in the frequency domain, a time domain realization is performed at run time. Considering a wave spectrum discretized into N individual harmonic components, each with amplitude ζ_n, frequency ω_n and phase θ_n, the first order seakeeping problem can be solved according to:

$$\boldsymbol{\tau}_{waves}(t) = \sum_{n=1}^{N} \boldsymbol{H}(\omega_n)\zeta_n \, cos(\omega_n t + \delta(\omega_n) + \theta_n),$$

$\boldsymbol{H}(\omega_n)$ and $\delta(\omega_n)$ are the 6-DOF force transfer function amplitude and phase as functions of wave encounter frequency ω_n, respectively, and t is time. For our case, the code solving this part of the input to the equations of motion is wrapped in the *Vessel FMU*, a Functional Mock-up Unit (FMU) publicly available in the OSP Reference models repository on GitHub, which is described in more detail by *Hassani et al. (2016), Skjong et al. (2018).*

2.2. Structural responses

While the seakeeping task is mainly associated with solving the equations of motion, shown in Eq.(1), the structural analysis concerns wave-induced loads such as tension-, and shear forces as well as torsional and bending moments at various transverse cuts along the longitudinal direction of the vessel. When a vessel operates at sea, it is continuously exposed to the dynamic forces generated by ocean waves. These interactions between the vessel and the surrounding wave environment result in wave-induced responses, which are essential to understand for ensuring safe, efficient, and cost-effective ship design and operation.

In addition to supporting design and engineering activities, such estimations can assist the crew during daily operations. Predicted bending and torsional loads across alternative vessel positions, headings and speeds enable route planning that minimizes structural stress in heavy seas. Information on local responses helps protect cargo and passengers by identifying conditions with high accelerations, while continuous stress monitoring increases awareness of fatigue for critical equipment. The simulator can also issue early warnings when approaching limits related to slamming loads or green water on deck. During challenging maneuvers, feedback on structural responses further supports safe speed and heading adjustments, reducing the risk of damage or injury for personnel and cargo.

To support this need, we again turn to the VERES program, *Fathi (2018), Fathi and Hoff (2017),* and use it as basis for calculating the structural responses based on the global wave-induced loads on the vessel. The program calculates global load transfer functions defined for a series of transverse cuts along the vessel. Like in Section 2.1, a wave spectrum discretization is employed to provide a time domain solution as follows:

$$\boldsymbol{V}(t) = \sum_{n=1}^{N} \boldsymbol{V_a}(\omega_n)\zeta_n \, cos(\omega_n t + \delta(\omega_n) + \theta_n).$$

$\boldsymbol{V_a}(\omega_n)$ and $\delta(\omega_n)$ are the 6-DOF global load transfer function amplitude and phase as functions of the wave encounter frequency ω_n, respectively. Unlike the seakeeping solver mentioned in Section 2.1, the wave-induced loads analysis is not part of the OSP reference model library. Hence, an extended version of the Vessel FMU has been developed as part of the EDINAF project and is used in this paper.

2.3. Coordinate transformations

In simulation-based maritime systems, different components may operate using distinct coordinate reference frames. A common example is the use of geodetic coordinates (latitude/longitude) versus local Cartesian frames such as the North-East-Down (NED) frame. Fig.2 illustrates the graphical relationship between these two frames. To ensure consistent spatial interpretation and accurate data exchange between such components, robust coordinate transformation mechanisms are essential within the OSP framework. The challenges associated with this issue are thoroughly discussed in *Sadjina et al. (2024).*

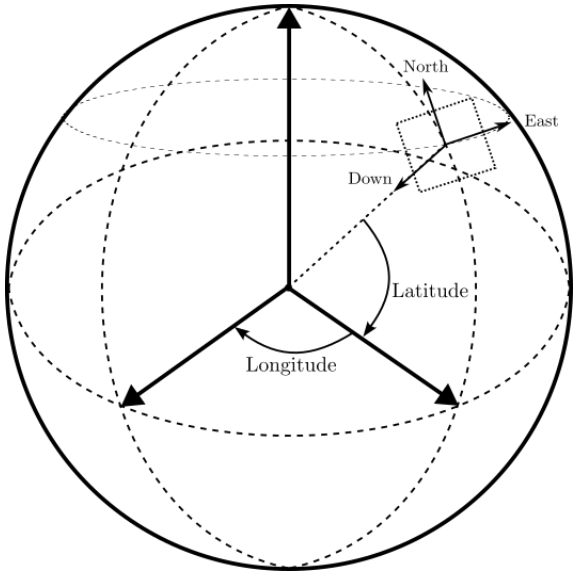


Fig.2: Geodetic coordinates (latitude/longitude) and North-East-Down frame

To address these challenges, we have developed dedicated FMUs for transforming between geodetic (WGS84) coordinates and local NED frames. These transformation FMUs operate strictly on position data: latitude, longitude, and altitude are converted to or from local north, east, and down coordinates. Other physical quantities such as velocities, accelerations, or forces are not transformed by these modules, since their frame dependence requires additional handling such as rotation of vectors according to the vehicle attitude. In the present work, this focus on position transformations alone is sufficient, as it ensures a consistent spatial reference for the models relevant to our use case, such as navigation, seakeeping, and sensor systems.

The transformations are implemented using the lla2ned and ned2lla functions provided by MATLAB, https://se.mathworks.com/help/nav/ref/ned2lla.html, https://se.mathworks.com/help/nav/ref/lla2ned.html, which are wrapped in FMU containers for use in OSP-compliant simulations. Notably, we apply the "flat" argument option in these functions, as it is better suited for the localized operational context of our use case. The required reference point (i.e., the local frame origin) is specified by the user and provided during simulation initialization.

In addition to geodetic-to-NED transformations, conversions between polar and Cartesian coordinates are also required. These likewise apply to position data only, and since they follow standard mathematical procedures, they are straightforward and do not require further elaboration here.

All transformation FMUs are implemented as self-contained modules and can be seamlessly integrated into OSP-based systems. This modularity ensures that existing FMUs do not need to be modified, supporting interoperability across a wide range of models and coordinate systems.

3. Communication

Communication between connected systems and the OSP-based simulation is typically handled via network FMUs, https://opensimulationplatform.com/use-cases/. These are FMUs that encapsulate network protocols, enabling external systems to use their preferred communication protocols while maintaining compatibility with the FMI-based simulation environment. By wrapping the communication protocol into an FMU, no additional support is required from the simulator core for specific interfaces.

The system integrator configures communication between network FMUs and model FMUs in the same way as connections between standard model FMUs. Different network FMUs are used for distinct interfaces or system types, allowing each to maintain its own protocol logic. This approach preserves the physical and logical topology of the system and promotes modular, reusable communication components. Furthermore, OSP-based simulations can be executed in real time, even when high-fidelity models are involved.

Several network FMUs are publicly available - such as DDS FMU and NMEA FMU. However, to the best of the authors' knowledge, an MQTT-based network FMU had not been available prior to this work, necessitating the development of a new MQTT FMU.

MQTT (Message Queuing Telemetry Transport) is a widely used protocol for lightweight, message-oriented communication. It follows a publish-subscribe model, making it especially suitable for applications involving constrained bandwidth, high-latency networks, and asynchronous data exchange. A key strength of MQTT is its support for bidirectional communication—an essential requirement in digital twin applications. This enables not only real-time data ingestion from a physical asset into the digital twin, but also the transmission of commands or control signals back to the real-world system. Such two-way interaction is critical for enabling synchronization, remote control, predictive maintenance, and real-time decision-making.

To support this, a generic MQTT FMU was developed as part of the EDINAF project. This FMU provides seamless integration of MQTT messaging into FMI-based co-simulations. It acts as a bridge between MQTT signals and FMI variables, allowing distributed tools to exchange data without requiring code compilation or direct protocol implementation within each model. Configuration is flexible and managed through external files defining MQTT topics, data types, and broker connection settings. The FMU supports multiple payload formats, including CSV and JSON, and generates the necessary "modelDescription.xml" to conform with the FMI standard.

This architecture makes the MQTT FMU particularly well-suited for real-time co-simulation and deployment in modular, distributed simulation environments. A schematic illustration of the MQTT FMU and its connections within a simulation context is shown in Fig.3.

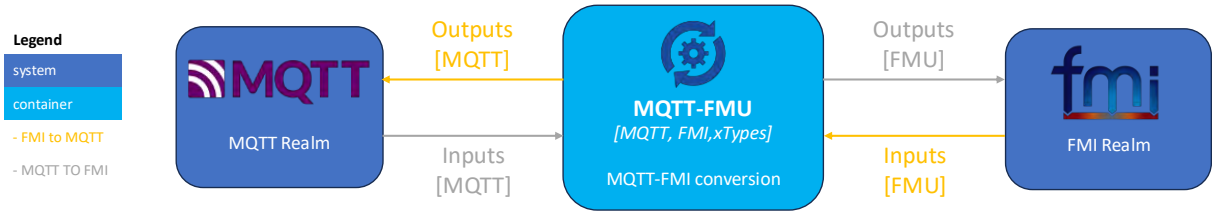


Fig.3: System overview of the MQTT FMU and its connection to the MQTT and FMI realms

4. Use case

To demonstrate some of the capabilities of the Open Simulation Platform (OSP) for onboard digital twin applications, we use the research vessel R/V Gunnerus as a case study. A photo of the vessel, along with its key specifications and dimensions, is shown in Fig.4. R/V Gunnerus is a 36 m long research vessel owned by the Norwegian University of Science and Technology (NTNU). The vessel is well equipped for research purposes and features a comprehensive suite of onboard sensors, including GNSS systems, as well as an MQTT server for streaming sensor data.

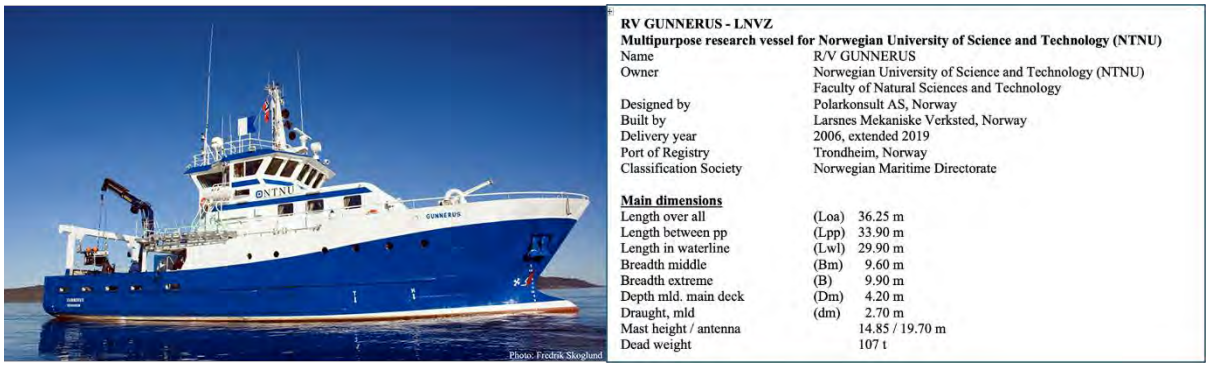


Fig.4: R/V Gunnerus with corresponding properties

An overview of the ship's IT system infrastructure is provided in Fig.5. This includes various sensors, onboard network configuration, and data distribution via MQTT.

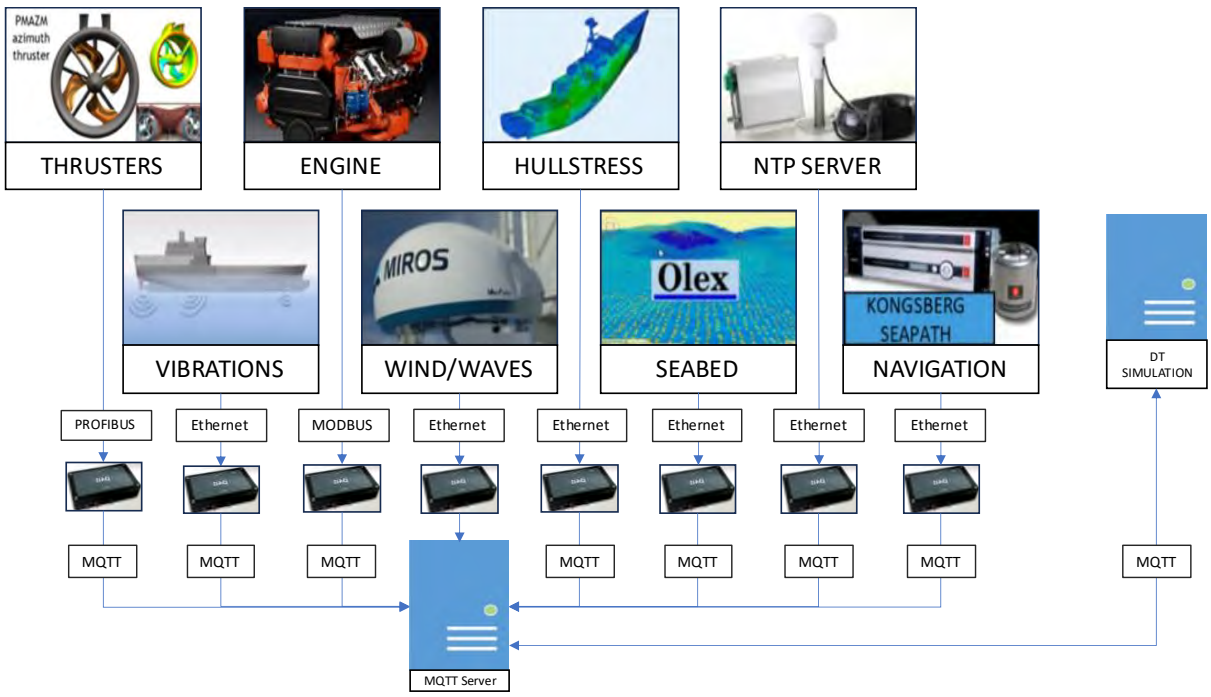


Fig.5: R/V Gunnerus IT system infrastructure

4.1. Software architecture

A digital twin (DT) has been deployed within an OSP instance running onboard Gunnerus. The digital twin comprises a seakeeping simulation model, which estimates ship motions, and a structural model, which calculates structural responses. This twin is connected in real time to live data streams from the vessel's sensors, enabling both data ingestion and result dissemination. Communication between the physical vessel and the simulation is facilitated using MQTT, allowing the digital twin to both receive data from the sensors and return simulation outputs to other onboard systems.

The implementation setup is illustrated in Fig.6. The digital twin requires input from the following onboard sensors: GNSS, port and starboard thruster systems, wave radar, and wind sensor. To integrate live sensor data into the OSP simulation, a communication unit must be included. In this case, an MQTT FMU is used, as the onboard MQTT server provides sensor data via publish-subscribe messaging.

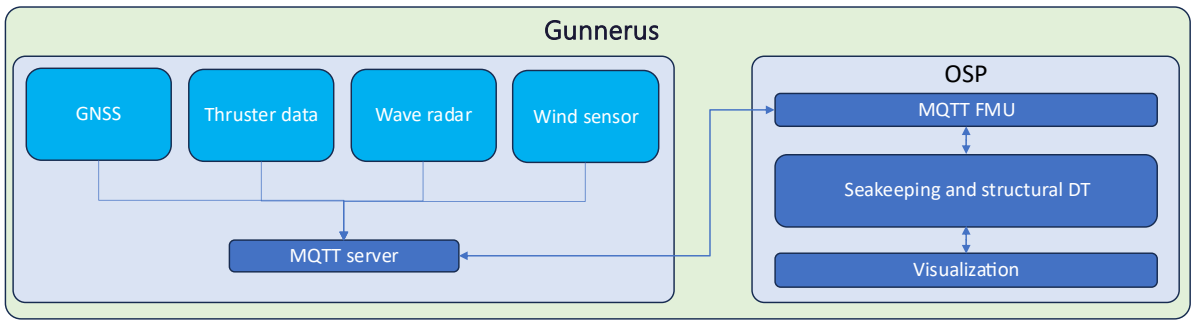


Fig.6: OSP running digital twin onboard R/V Gunnerus

4.2. MQTT communication

As shown in the left part of Fig.6, the MQTT server collects data from the individual sensors. Each sensor is typically associated with a dedicated application that extracts the data and publishes it to the vessel's MQTT server. For security reasons, this server is restricted to the onboard network.

Data exchange via MQTT follows a publish-subscribe model. Table I provides an overview of the mapping between each sensor and its associated MQTT topic used in the seakeeping digital twin. Notably, Gunnerus is equipped with two main thrusters, one on the port side and one on the starboard side.

Table I: Mapping between sensors and MQTT topics

Sensor	**MQTT topic**	**Data**
GNSS	Gunnerus/demo/Seapath	Latitude position Longitude position Heading
Port thruster data	Gunnerus/hcx_port_mp	RPM percentage Rudder Angle
Starboard thruster data	Gunnerus/hcx_stbd_mp	RPM percentage Rudder Angle
Wave radar	Gunnerus/WaveRadar	Current speed (north/east) Wave period Significant wave height Wave direction
Wind sensor	Gunnerus/dpWind	Wind direction Wind speed

In the right part of Fig.6, the OSP application layout is shown. The MQTT FMU is configured to connect to the onboard MQTT server using the correct credentials and topic mappings. This FMU extracts the required signals and feeds them into the OSP simulation. The digital twin used in this case study and highlighted in blue in Fig.8, consists of simulation models listed in Table II. Details about the various FMUs of Fig.8 are listed in Table II. The entire OSP application runs on a server with a Mint Linux operating system onboard Gunnerus.

4.3. Visualization

To enable real-time visualization, a customized version of the Cosim DemoApp is used, as illustrated in the lower right corner of Fig.6, with a screenshot provided in Fig.7. Beyond visualization, the DemoApp records log files (.log and .csv), offering valuable data for post-simulation analysis and operational insight. It also supports real-time adjustment of OSP values and parameters, allowing users to actively influence the simulation, a functionality depicted by the two-way arrow linked to the Visualization block in Fig.6.

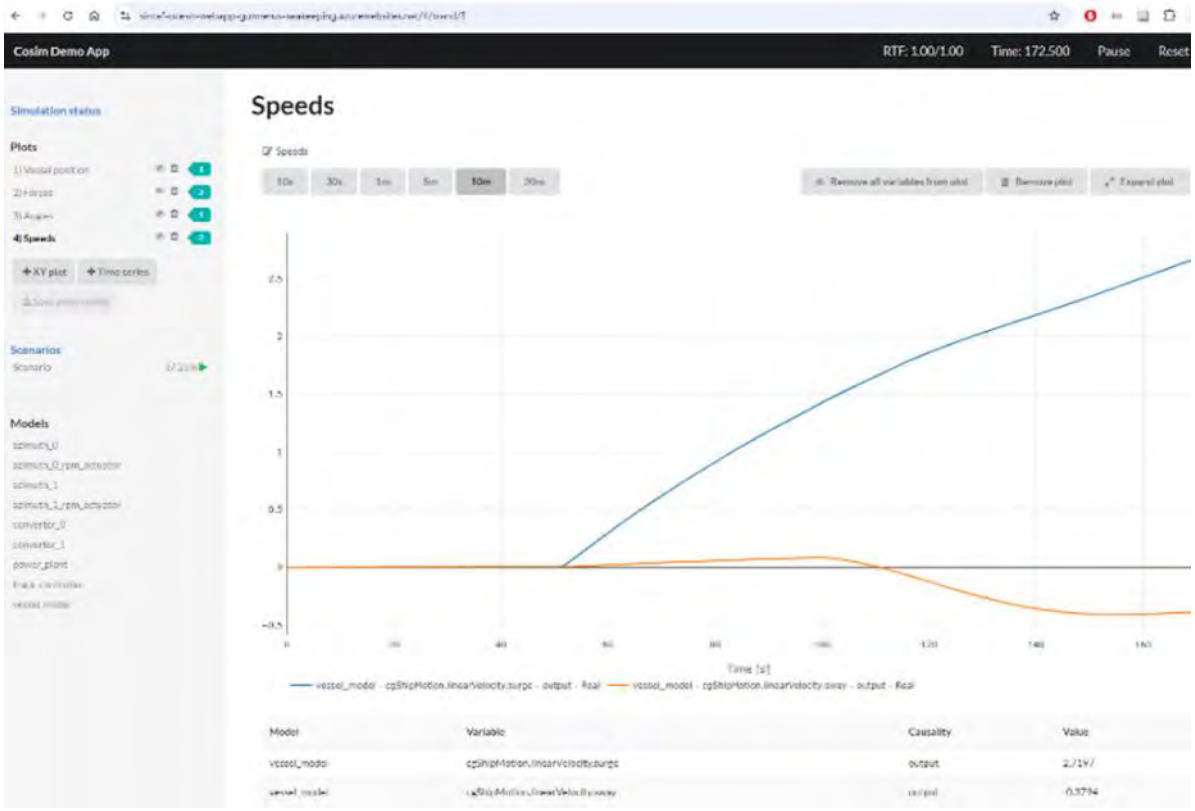


Fig.7: Screenshot of Cosim DemoApp, illustrating states of the Gunnerus DT

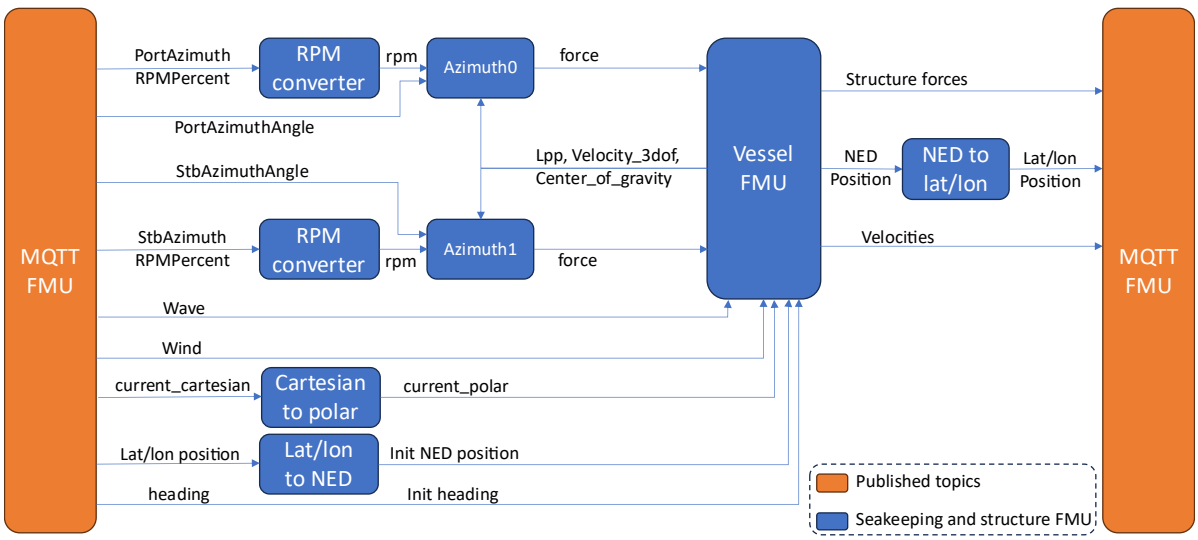


Fig.8: Dataflow between Gunnerus MQTT topics and OSP digital twin

Table II: Overview of FMUs used in the OSP simulation

FMU name	Description	Modelling tool
Cartesian to polar converter	Converts from cartesian (north/east vectors) to polar coordinates (angle and magnitude)	MATLAB/Simulink
Lat/lon to NED	Convert latitude/longitude to North-east-down (NED) coordinates.	MATLAB/Simulink
NED to lat/lon	Convert North-east-down (NED) coordinates to latitude/longitude.	MATLAB/Simulink
RPM converter	Converts RPM feedback given in % of max RPM to true RPM	C++
Azimuth0 and Azimuth1	Main propulsion azimuth thruster. Computes the force generated from the thrusters.	VERES, *Fathi (2018)*
Vessel FMU	6DOF hull model including structure analysis and environment model. Computes the position, velocity, and center of gravity of Gunnerus as well as wave-induced loads.	VERES, *Fathi (2018)*

4.4 Test campaign

Testing of the R/V Gunnerus digital twin formed part of the live Integrated Use Case demonstration held in Trondheim during the fifth General Assembly of the EDINAF project, from 20 to 22 May 2025. Alongside the digital twin trials, the event featured battle simulations, resilience testing, and cyber-attack response exercises. Services and equipment were distributed across Trondheim harbour and the Trondheim fjord, creating a large, integrated test environment. Participants also observed notable advances in digital twinning technologies and decision-support tools for autonomous systems. The demonstration involved multiple vessels and land-based services operating in coordination, with the R/V Gunnerus track shown in Fig.9.

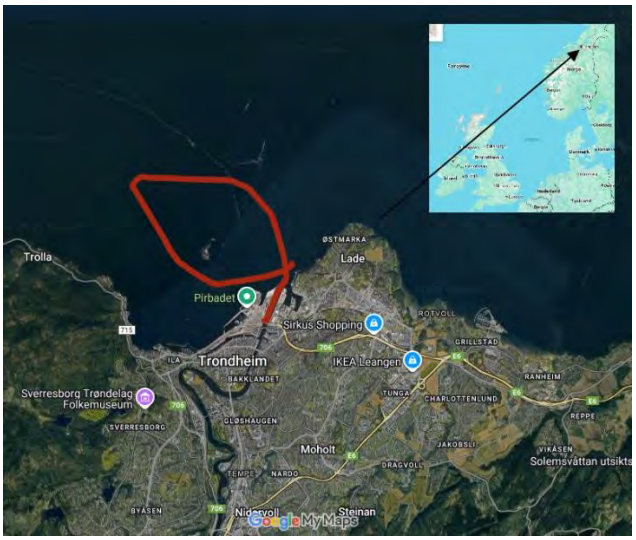


Fig.9: Test track of R/V Gunnerus

5. Discussion

The proof-of-concept implementation presented in this work demonstrates that the Open Simulation Platform can be effectively applied to real-time, two-way communication between a physical vessel and a digital twin. By combining existing OSP reference models with newly developed components, i.e. the MQTT FMU, we have shown that live sensor data can be integrated, processed, and linked with simulation models to create estimates of vessel motions and structural responses. The system performed reliably during onboard testing, confirming the feasibility of standards-based communication for distributed maritime simulations.

From a technical perspective, the decision to implement MQTT as the primary communication mechanism was successful, especially since the MQTT protocol was heavily utilized among connecting systems as well. MQTT's publish–subscribe architecture simplified the integration of sensor data streams and enabled modular connection between the physical vessel and simulation environment. This approach also provided flexibility for scaling the system to additional data sources and simulation components without significant changes to the communication layer. Compared to other evaluated protocols, MQTT offered the most straightforward path to achieving low-latency bidirectional exchange in a bandwidth-constrained maritime network environment.

The use of OSP reference models provided a robust starting point, reducing development time and ensuring consistency with FMI standards. However, this work also highlights a recurring challenge: available reference models often require extension to address specific operational needs. In our case, additional FMUs for coordinate transformation and structural load estimation were necessary to develop to achieve the desired functionality. This reinforces the importance of maintaining open, model libraries that support both general maritime applications and specialized operational needs.

The presented approach also has broader implications for operational decision support. By providing real-time estimates of vessel motions and structural responses, the digital twin can inform operators about potential comfort, safety, or structural limits before these are exceeded. While this work focused primarily on proof-of-concept demonstration, the same architecture could be extended to support predictive maintenance, autonomous control strategies, and cross-vessel coordination in fleet operations.

Finally, the study underlines the importance of interoperability and standards compliance in the maritime domain. The combination of FMI-based models, open communication protocols, and modular simulation design lowers the barrier to integrating digital twins with both legacy and new systems. As adoption of such architectures grows, further work will be needed to address cybersecurity, long-term model validation, and integration with shore-based decision support services.

6. Conclusion

A proof-of-concept digital twin for the R/V Gunnerus ship was developed using the Open Simulation Platform and extended with an MQTT-based communication FMU for real-time two-way communication and integration of live sensor data. Onboard trials confirmed that the modular, standards-compliant architecture enables prediction of vessel motions and structural responses in real time, within constraints of onboard computing resources, network bandwidth, sensor fidelity, and the vessel's operational capabilities. The results demonstrate the suitability of open, lightweight communication protocols and FMI-based models for scalable maritime digital twin applications, while future work will target improved model fidelity, broader interoperability, and secure long-term deployment.

Acknowledgements

This work has been financed in part by the Research Council of Norway (RCN) under the project SEACo: Safer, Easier, and more Accurate Co-simulations (RCN grant no. 326710, 2021–2025) and by the European Defence Fund (EDF) under grant agreement 101103273-EDINAF-EDF-2021-NAVAL-R-2, for which we are very grateful.

References

FATHI, D.E. (2018), *ShipX Vessel Responses (VERES), User's Manual*, SINTEF Ocean

FATHI, D.E.; HOFF, J.R. (2017), *ShipX Vessel Responses (VERES), Theory Manual*, SINTEF Ocean

FOSSEN, T.I. (2021), *Handbook of Marine Craft Hydrodynamics and Motion Control*, Wiley

HASSANI, V.; RINDARØY, M.; KYLLINGSTAD, L.T.; NIELSEN, J.B.; SADJINA, S.S.;

SKJONG, S.; FATHI, D.; JOHNSEN, T.; ÆSØY, V.; PEDERSEN, E. (2016), *Virtual Prototyping of Maritime Systems and Operations*, 35th Int. Conf. Ocean, Offshore and Arctic Engineering, Busan

SADJINA, S.; WILLE, E.; JØRGENSEN, U.; SKJONG, S. (2024), *Coordinate Transformation Techniques for Improved Co-Simulation in the Maritime Industry*, J. Physics Conf. Series 2867, https://doi.org/10.1088/1742-6596/2867/1/012052

SALVESEN, N.; TUCK, E.O.; FALTINSEN, O. (1970), *Ship motions and sea loads*, SciSpace - Paper 78, pp.250–287

SKJONG, S.; RINDARØY, M.; KYLLINGSTAD, L.T.; ÆSØY, V.; PEDERSEN, E. (2018), *Virtual prototyping of maritime systems and operations: applications of distributed co-simulations*, J. Mar. Sci. Technol. 23, pp.835–853

A 3D Digital Twin for Monitoring the Hull Condition and AI For Automating Detection of Defects

Erik Stensrud, DNV, Oslo/Norway, erik.stensrud@dnv.com
Xiao Liang Gong, DNV, Shanghai/China, Xiao.Liang.Shandy.Gong@dnv.com
Geir Hamre, DNV, Oslo/Norway, geir.hamre@dnv.com
Darshana Abeyrathna Kuruge, DNV, Oslo/Norway, Darshana.Abeyrathna.Kuruge@dnv.com
Yi Edward Liu, DNV, Oslo/Norway, Yi.Edward.Liu@dnv.com
Bahman Raeissi, DNV, Oslo/Norway, Bahman.raeissi@dnv.com
Torbjørn Skramstad, NTNU, Trondheim/Norway, torbjorn.skramstad@ntnu.no
Vetle Nordang, NTNU, Trondheim/Norway, nordang.vetle@gmail.com
Daniella Nguyen, NTNU, Gjøvik/Norway, Daniella_nguyen@hotmail.com

Abstract

This study reports status on a) the development of a 3D-based digital twin for monitoring the hull condition, b) automated recognition of structural elements of the hull in point clouds, c) automated image quality assessment, d) updates on crack and corrosion detection performance in images, and e) automated detection and measurement of deformations and pitting in point clouds. Previous COMPIT papers have reported as follows: i) a concept for a continuous, digital, remote hull survey process, ii) results on automated corrosion and crack detection, and iii) results on automated detection of deformations in images and then measurement of the deformations with depth cameras.

1. Introduction

The paper provides an update on R&D results since COMPIT 2023. The motivation and background of this R&D project and initial results were reported in previous COMPIT papers. The technical feasibility of drone-based inspections was investigated in *Stensrud et al. (2019)*. Updated results were reported in *Stensrud et al. (2020)* and *Stensrud and Klausen (2022)* whereas *Stensrud et al. (2023)* reported a concept of a new, digital process for hull surveys that takes advantage of robot technologies to collect data of the physical asset; furthermore, AI algorithms to detect cracks and deformations in images; detection of geometric defects such as corrosion scabs, in high-precision point clouds; and initial trials, using a lab sample, on detecting and measuring geometric defects such as buckling, in point clouds collected from a RGB-D camera.

2. Automatically creating a geometric 3D digital twin from point clouds exploiting parametric models

Arguably, 3D models improve hull integrity management, and 3D visualisations of geometric digital twins improve the monitoring of the asset condition. At present, few assets have 3D digital twins but rather rely on 2D drawings for approval, and documents for keeping track of the hull condition. Therefore, 3D models must be created for existing assets, and it is of interest to find cheap solutions.

Ship Manager Hull probably "… is the most advanced hull integrity management software available - for maintenance strategy, planning, inspections, hull surveys, assessments and documentation," https://www.dnv.com/services/hull-integrity-and-ship-maintenance-software-shipmanager-hull-1531/. Ship Manager Hull has, however, a limited customer base partly because the 3D hull model is vessel-specific and manually created for each individual vessel from 2D drawings and therefore can be (too) expensive.

In the REDHUS project (grant number 317773, Research Council of Norway), motivated by the need for cheaper 3D models and automatic inspection planning for drones, it was realized that data collected by drones could be used to create 3D structural digital twins. The first investigation converted point clouds collected by a stationary laser scanner into a 3D mesh model, depicted in

Fig.1. This approach required some manual work and is reported in *Stensrud and Bjørdalsbakke (2023)*. In this approach, structural elements within the compartment are not recognised, i.e., there is no semantics, no "understanding" of what the robot looks at. This "understanding" is deemed crucial for close-up inspections to ensure the robot inspects the structural element targeted for inspection.

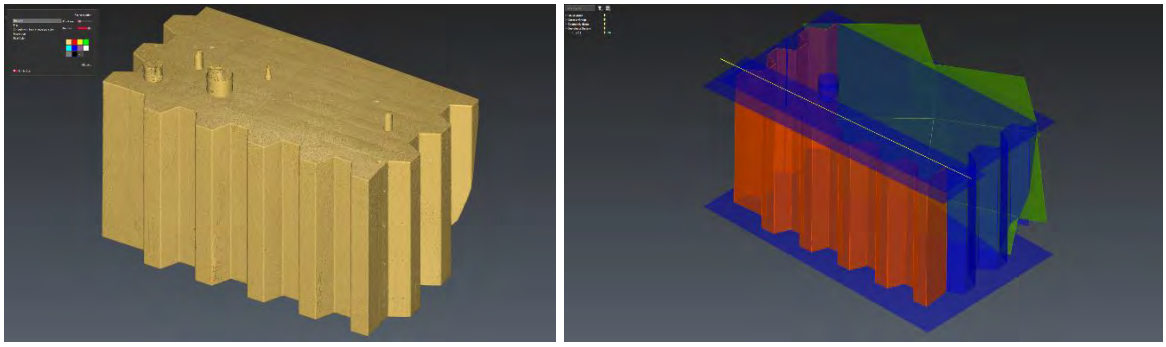


Fig.1: Smooth point cloud (left); 3D geometric model from point cloud (right); *Stensrud and Bjørdalsbakke (2023)*

Next, it was realized that large parts of the vessel hull structure are built according to parametric models. In this context, a parametric model is a spatial structure with a repeating pattern of a limited number of different structural element types, interconnected and organised hierarchically, Fig.2. For example, a cargo compartment is composed of structural elements like web frames and deck transverses, and each of these structural elements can be further broken down into elements such as girders and brackets, and so on. Each structural element type such as a web frame can be described by parameters such as its dimensions and shape, and the compartment can be described with parameters such as its number of web frames. For ballast water tanks, the compartment sections often are structurally the same and therefore have the same parameters (same compartment size, same number of repeating elements, same shape of the components of a given structural element type, Fig.3.

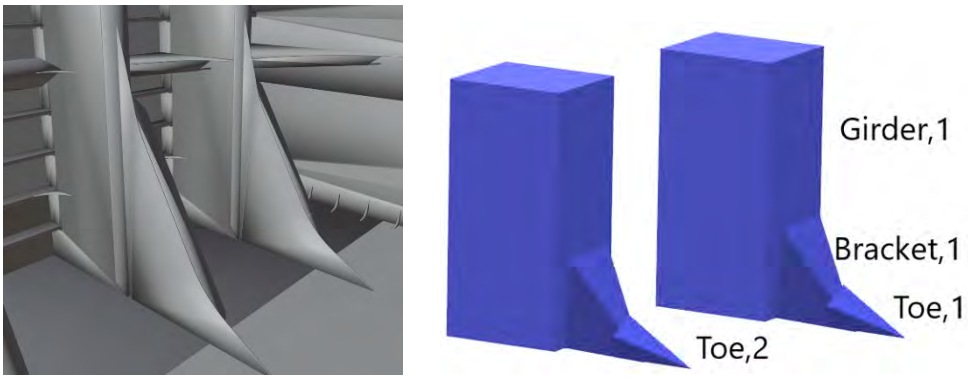


Fig.2: Example of a parametric model of a structural element type

The parameters of the parametric model can be established through a bottom-up approach using a semantic scene graph and identifying known structural components in the collected data. The suggested method can enable the creation of a 3D model of the vessel hull with semantic information. This would then also enable automatic inspection planning, in particular close-up examination, based on the additional knowledge of how the different semantic elements (structural element types) should be inspected.

The insight on parametric hull structures was exploited by *Dharmadhikari and Alexis (2025)* in their novel semantic inspection paradigm applied to their predictive path planning algorithm. The autonomous drone identifies structural elements inside a compartment and generates a semantic scene graph. This "understanding" of compartment semantics is used to perform targeted close-up examination. Their results demonstrated that the flight time is significantly reduced compared to state-of-the-art methods.

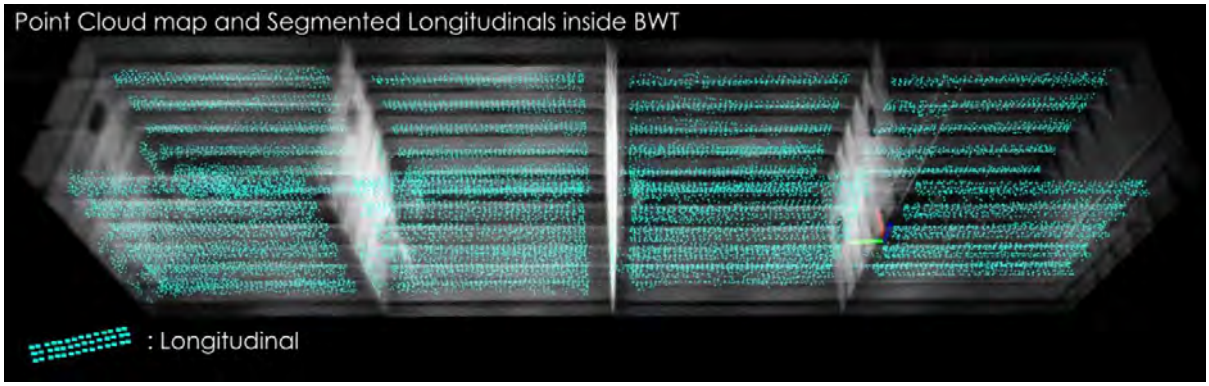


Fig.3: Autonomous drone mapping the next BWT section reusing mapping knowledge of the previous section. (courtesy by Mihir Dharmadhikari)

3. Monitoring the asset condition with a 3D geometric digital twin

The development effort on the digital twin has focused on making the digital twin more production-ready since COMPIT 2023, e.g., automate the execution of algorithms in the digital twin's back end, improve scalability and processing speed, and ensure the same workflow irrespective of whether to run the corrosion detector, crack detector, or deformations detector. Other updates include improved vessel management and user management, for example importing a vessel from the production system or adding a vessel in the application or granting access to external users.

Integration with a flight planning module of an external drone provider has also been developed. In addition, the hit rates of crack and corrosion detectors have been improved.

User interface and data model were modified and improved. Earlier versions were survey centric whereas it now is asset centric, Fig.4. In the survey centric twin, "Survey" was the top-level entity type. It was then cumbersome to find related, historic surveys of the same compartment, such as previous findings (cracks and corrosion) of a specific compartment. The asset centric digital twin facilitates finding the right ship and then drilling down (top-down navigation) to the right compartment to identify historic findings of that compartment. A zoom function improves visualization of video frames, Fig.5.

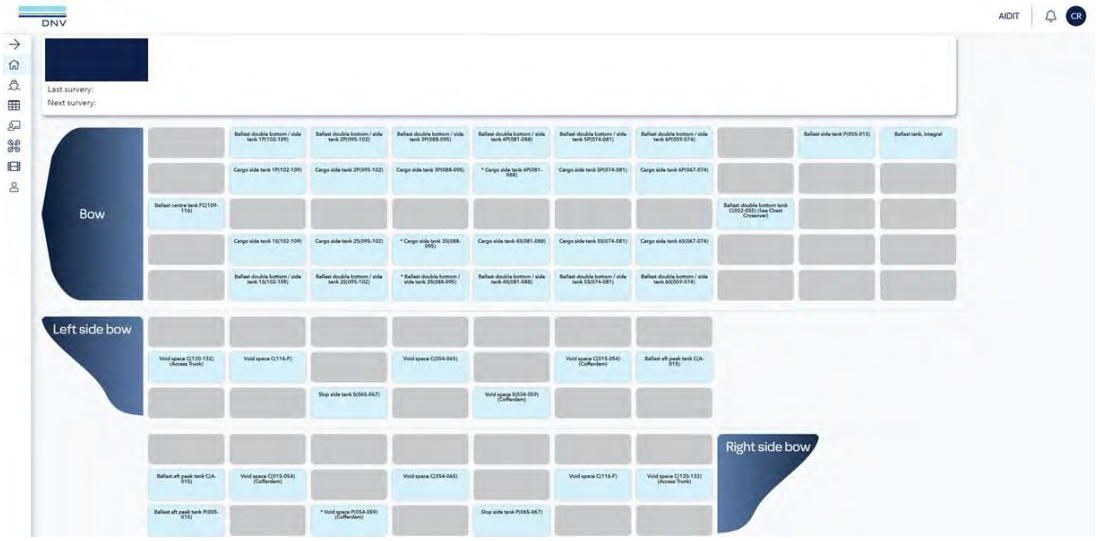


Fig.4: Illustration of the asset centric view of the digital twin

Fig.5: Zoom function for improved visualization of video frames

Inspection planning has been improved, for example functionality to manage inspection points such as adding or modifying them. The data types of a compartment include point clouds in addition to videos 3D model. Findings are location-tagged and displayed in both video and the 3D model. Also, drone inspection data from external drone providers can be imported. including drone locations.

Fig.6 shows an updated version of the video inspection tool user interface since *Stensrud et al. (2022)*. The drone position in the compartment is displayed in the 3D model and is synchronized with the video shown in the video viewer. The functionality has been improved since COMPIT 2022. Back then, a single bottom bar highlighted the locations of video frames where the AI had detected anomalies (cracks or corrosion). In the current version, there are four bottom bars, cracks detected by the AI (blue), corrosion detected by the AI (red), manually added findings (yellow), and manually added annotations other than findings on a video frame (green), respectively. The functionality enables efficient review of the video and the location-tagged findings detected by the AI crack and corrosion detection algorithms. A finding may be confirmed, deleted modified, or manually added in the digital twin by drawing over the video, Fig.6. (The view perspectives of video and 3D model don't necessarily match, but the finding location does in the two views.) Findings are indicated in the 3D model with yellow dots, and a finding can be selected (then displayed as a pink dot) in the 3D model to display information about it, Fig.7.

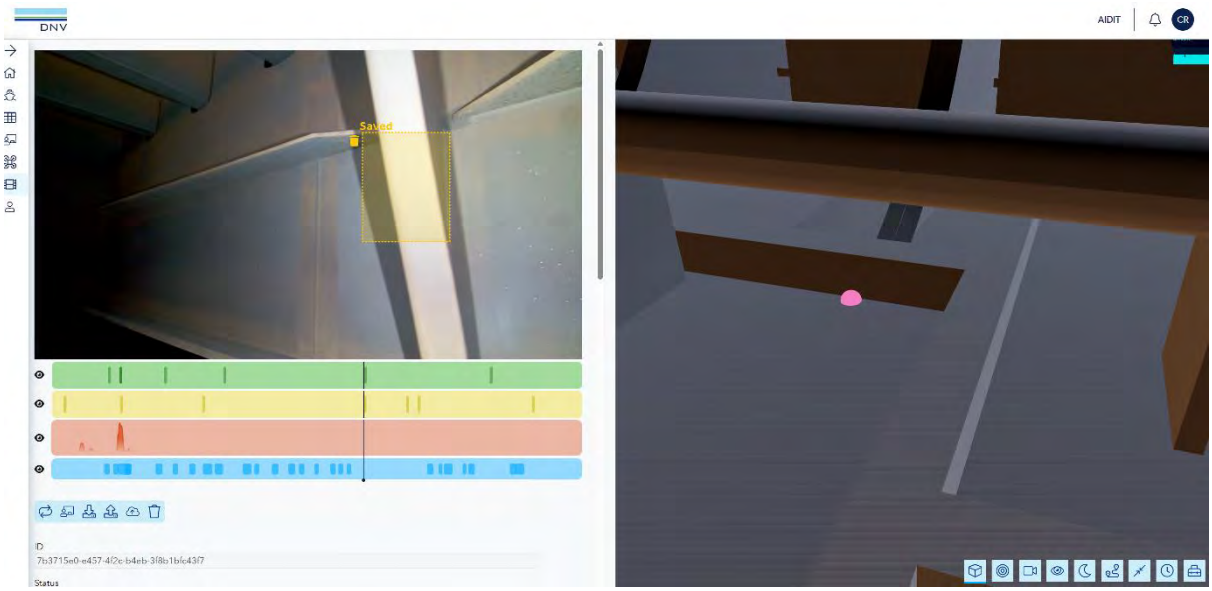


Fig.6: Adding a finding manually by drawing over the video (left) and displaying its location in the 3D model (right)

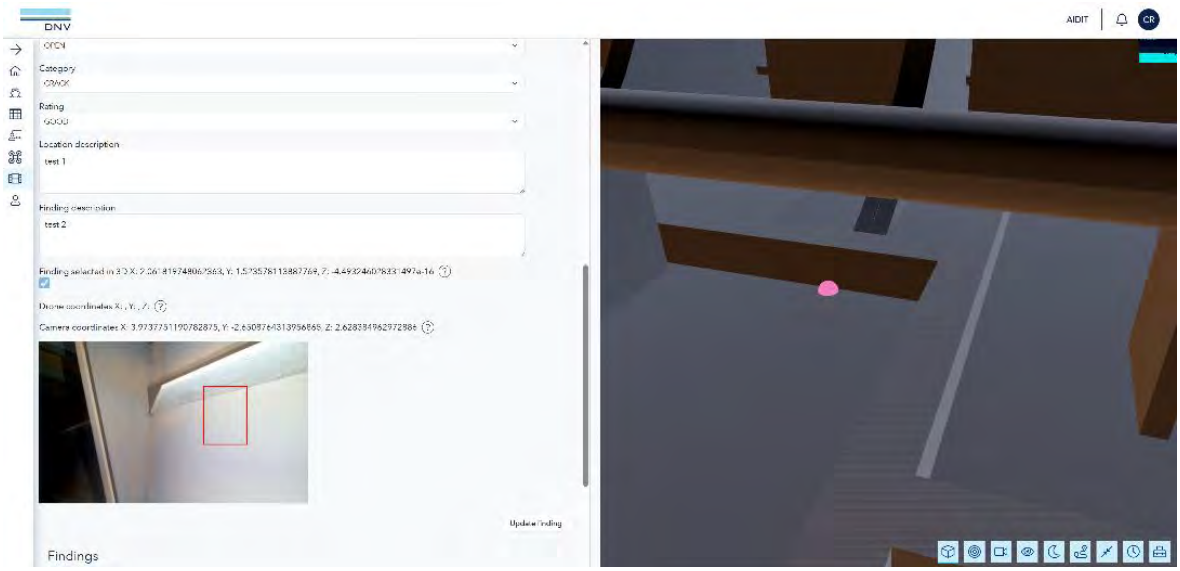


Fig.7: Displaying information about a finding selected in the 3D model

4. Automated detection and measurement of deformations and pitting in point clouds

Deformations may be detected and measured from point clouds either by comparing them with a 3D model or by assuming or knowing the geometry of the surface. An algorithm has been developed to detect deformation, such as indent in plates, automatically from point clouds. The requirements in the algorithm for detection of indents are implemented according to our guidelines. For indents in plates, the maximum allowable indent depth is given by the formula in Fig.8. The spacing S between stiffeners typically is in the 600-800 mm range. Therefore, the maximum allowable depth δ typically is in the 5-6 cm range.

$$\delta \leq \frac{S}{12}(mm)$$

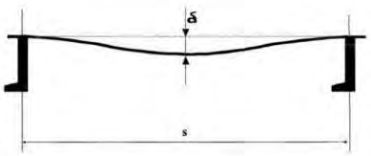


Fig.8: Maximum allowable indent in plates

A point cloud of a plate in the size of 4x6 m^2 that has five spacings between stiffeners has been generated for testing the algorithm, Fig.9, left. An artificial indent has been introduced in the middle of the plate, Fig.9, middle, with a peak of 10 cm. The spacing between stiffeners is 80 cm. The algorithm successfully detects the area where the indent exceeds the threshold δ and highlights it as a deformation, Fig.9, right.

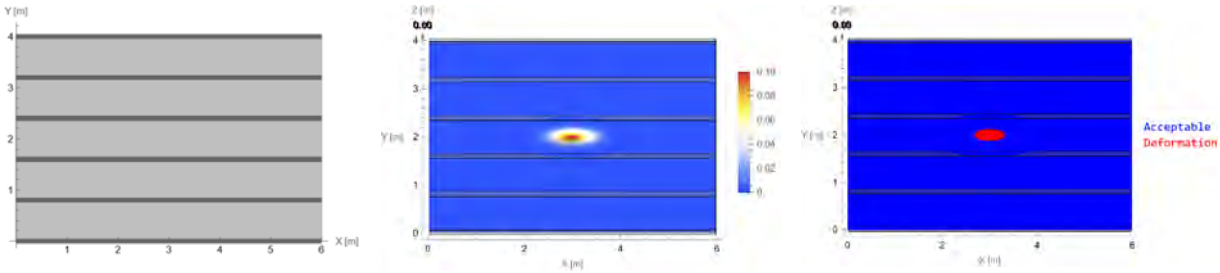


Fig.9: Indent detection: Generated point clouds of a surface with five stiffener spacings (left). Generated indent in the middle of the plate (middle). Detected deformation by algorithm (right).

Another requirement for detecting and reporting indent is whether the indent area of a plate has affected a region that is exceeding three spacings (3*S*). This allowable extent of indents has been implemented in the algorithm. Fig.10 depicts the results from a large indent that was introduced in the point cloud data (left), and the algorithm was able to automatically highlight the deformation area, both the large region exceeding 3*S* and the area above the threshold δ.

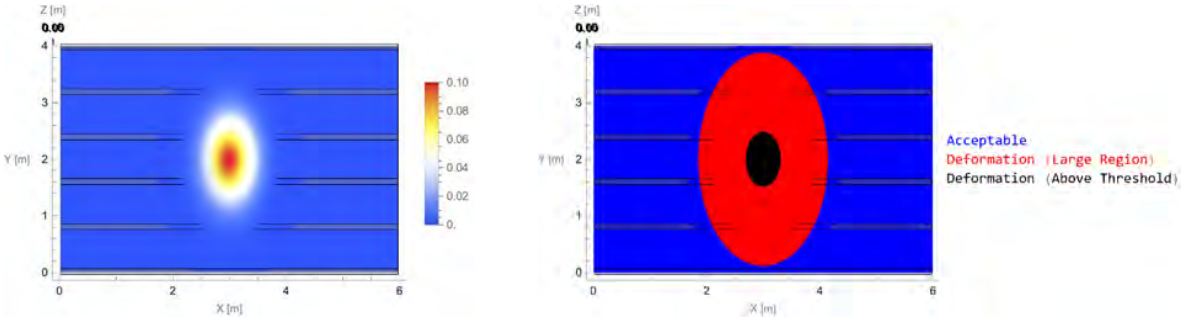


Fig.10: Large indent detection: large indent exceeding 3 x spacing (left). Detected deformation area by algorithm (right).

Fig.11 (left) shows the back-wall of a mock-up tank at the testing facility of University of Genoa, *Poggi et al. (2020)*. The wall contains pitting and was scanned by a high precision stationary laser scanner. The selected area of the scene for further analysis is marked by a red rectangle. The point cloud colors show deviations from a flat surface where the dark blue area is the flat reference. The other colors indicate deviations from the plane, either indents or bumps. The deviations from a completely flat plane might have been caused by the structures on the back of the plate such as welded stiffeners or in the production of the plate, Fig.11 (middle). Fig.11 (right) shows the pitting and other deviations, measured in mm. Apart from the pitting, the deviations are less than 5-10 mm. Such small anomalies would not have been easily detected by the naked eye. The pitting is measurable in the point cloud and are up to 15 mm deep but do not penetrate the plate thickness.

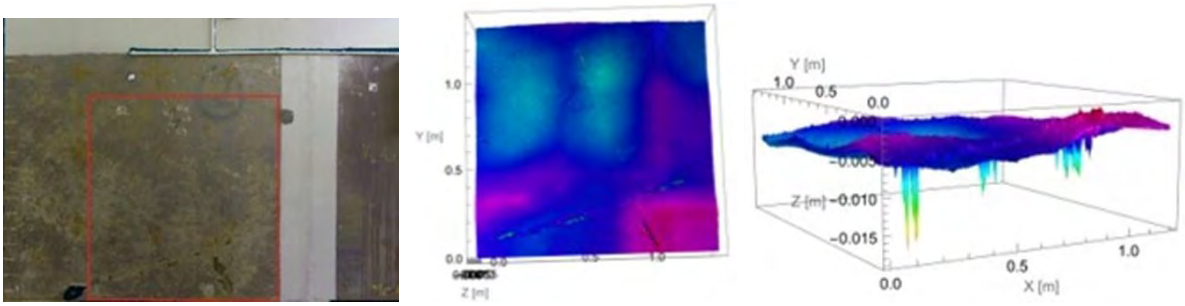


Fig.11: Pitting detection: the mock-up wall where the red rectangle is the area shown in the point cloud (left); point cloud of the selected area (middle). The same point cloud, showing the pitting and their depths (right).

5. Image quality assessment of close-up images containing cracks

Robots collect large amounts of images and video, often of varying quality. It would be useful for the inspector to know which images are good and which are bad to save time sifting through large amounts of bad images. Image Quality Metrics (IQM) may potentially be used to sort images into "good" and "bad" categories. To our knowledge, IQMs are mostly used to assess the quality of the whole image which might be a challenge for close-up photos inside confined spaces that may include cracks. These close-up photos exhibit complex, mixed scenes with respect to metrics such as blur, exposure, etc. due to the complex 3D environment and uneven lighting conditions. Some areas of the image might be very underexposed and blurry whereas other areas may be sharp and with correct exposure, as illustrated in Fig.12. The global IQM values for such a close-up image are not obvious, and therefore this was investigated.

Fig.12: Example of close-up image of a crack. The crack is in the foreground, in sharp focus and neither over- nor under-exposed. The upper left of the image is however underexposed (pitch dark) and out of focus (blurry).

A dataset of almost 4000 close-up images of cracks in coating and steel were subjectively categorized into “good” and “bad”, mainly based on how visible and certain the crack is to a human. Two persons reviewed the images together and discussed and agreed on the category based on agreed guidelines. (The guidelines include subjective certainty that the image contains a crack, that it is easy to label the crack correctly, that the image quality is sufficient to clearly identify the crack and understand what one is looking at.) The classification resulted in 70% good images and 30% bad images. A random subset of the good images was selected to achieve a 50/50 distribution of good and bad images for evaluating the IQMs.

The IQMs selected included exposure, blur, contrast, and salt and pepper noise. These were selected because these metrics seem most relevant for images of cracks since cracks are mostly dark pixels in contrast to the surrounding surface which often is lighter, especially in a coated ballast water compartment. For example, an underexposed image may not provide sufficient contrast to detect the dark crack. Likewise, a blurred image might mask the crack pixels since the number of crack pixels in an image is typically only 1% of the total pixels. The same argument applies to noise.

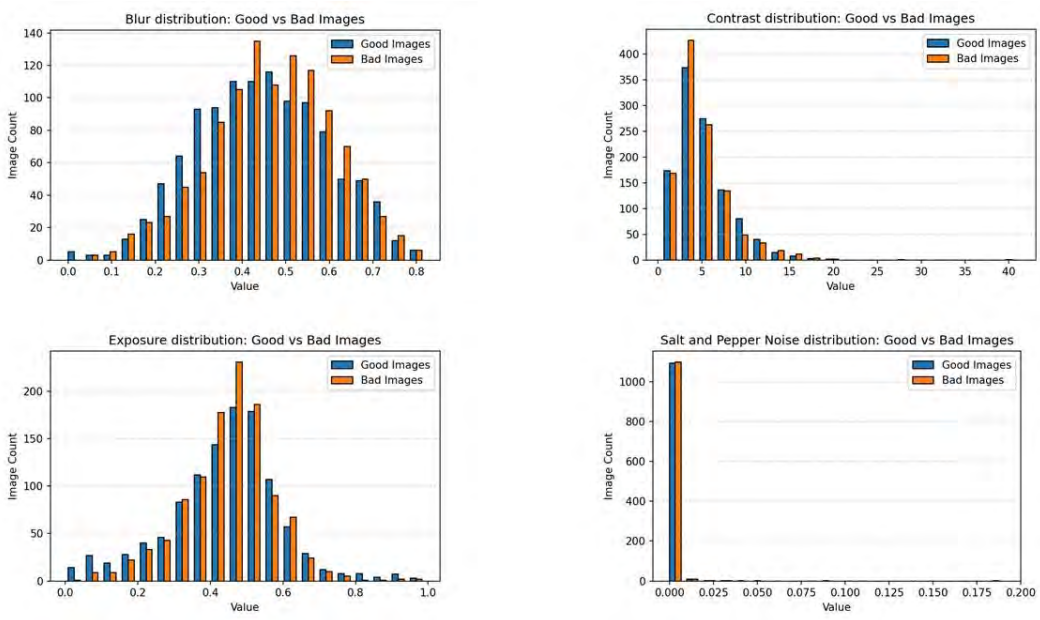


Fig.13: Histograms of good and bad images for blur, contrast, exposure, noise

The results suggest that these four IQMs only marginally manage to distinguish between the good and bad close-up images. Fig.13 shows the histograms for the four metrics of good and bad images. The value ranges are as follows: blur 0-1 where zero indicates no blur; contrast, values <1 indicate good contrast; exposure measures the overall exposure level (dark, bright, balanced), and the value range is 0-1 where zero indicates black; salt and pepper noise ranges from 0-1 where 0 is no noise.

The histograms exhibit substantial overlaps between bad and good for all four metrics. 'Blur' ranges from little blur to much blur with most of the images being half-blurry. Good images exhibit somewhat less blur than the bad. 'Contrast' is bad for most images, and good images do not seem to exhibit better contrast than bad images. 'Exposure' levels range from too dark to too bright. The majority of the images are reasonably balanced, and there is a slight overweight of bad images with balanced exposure. Salt or pepper noise seems almost absent and identical for good and bad images.

A classification and regression tree (CART) was also implemented to investigate whether a multi-dimensional metric would improve matters. One motivation for trying CART is that it is explainable, unlike deep neural networks. The 50/50 dataset was divided into 80% training data and 20% test set. The test set thus comprised 231 "bad" and 231 "good" images.

The classification tree and the results of the confusion matrix are shown in Fig.14. CART classified the majority as "bad". An interesting, and discomforting, result is that 75% of the "good" images were classified as "bad" (172 out of 231 "good" images). A more comforting result was that 77% (177 out of 231) "bad" images were classified as bad.

One possible explanation for the results could be that the metrics used overlap more clearly for "bad" images than for "good" ones, which makes the model more confident in classifying images as "bad." Another factor could be that the chosen metrics capture "badness" (such as blur or low contrast) more strongly than they capture "goodness." It is also possible that subjectivity in labeling or noise in the data made "good" images harder to learn. Finally, the simplicity of the CART model itself could limit its ability to capture the complexity of what makes an image "good." Taken together, these reasons could explain why the model predicts more images as "bad," but since its overall accuracy is close to 50%, we cannot rely on it for dependable classification. Finally, close-up images of cracks probably have very varying global image quality, see e.g. Fig.12, whereas the difference between a "good" and "bad" crack image depends on the local image quality, i.e. the quality in the area where the crack is.

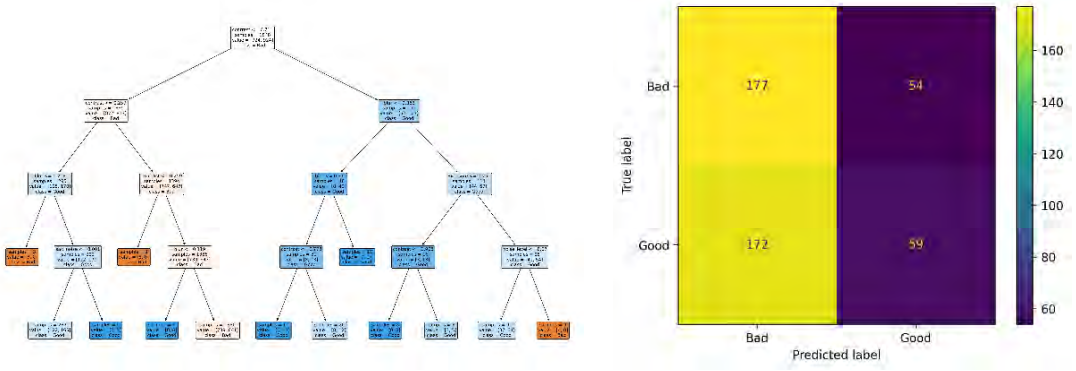


Fig.14: CART and confusion matrix

We also tried out a deep learning based blind image quality assessment (BIQA) method, *Zhang et al. (2023)*, on general images of cargo and ballast water tanks. (The term "blind" means the method evaluates an image's quality without having access to a pristine, distortion-free reference image for comparison, unlike Full-Reference methods that compare a distorted image to an original.) There is a number of BIQA methods including hand-crafted methods such as metrics-based; deep-learned approaches such as supervised- and unsupervised learning-based; multimodal quality assessment, and representative databases, *Wang (2023)*. BIQA methods remain, however, to be investigated on our specific dataset of close-up images of cracks.

6. Crack detection

Crack detection performance has been improved by introducing new deep learning models and additional techniques. A more advanced, recent deep learning model, YOLOV6 v3, has replaced the previous deep learning model, *Li et al. (2022).* The previous model had lower sensitivity to True Positives due to the disturbance of image quality, especially blur and brightness. To address this issue, the training data has been improved by adding more data augmentation techniques such as blur and brightness adjustment. During the training of YOLOv6 for object detection, the first five layers of the backbone were frozen. This aims to reduce overfitting to the highly homogeneous training data and encourage the learning of more generalized feature representations, thereby improving model robustness and performance on unseen datasets. Object tracking algorithms, specifically BOT-Sort, *Aharon et al. (2022),* and Byte-Track, *Zhang et al. (2022),* have enhanced video data inference performance. These algorithms enhance detection performance by leveraging cross-frame information.

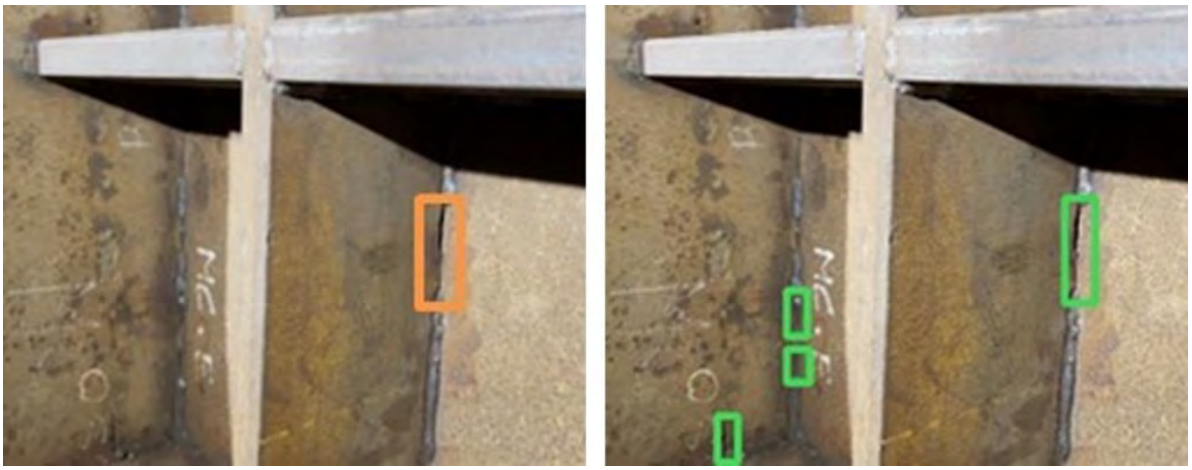


Fig.15: Example of improved crack detection: old deep learning model (left); current model and data (right)

The results exhibit an improved hit rate from 28% mAP50 to 35% mAP50 since *Stensrud et al. (2023).* (mAP50 = mean Average Precision with Intersection over Union threshold = 50%, https://en.wikipedia.org/wiki/Evaluation_measures_(information_retrieval).) The improved crack detection is illustrated in Fig.15. The improved accuracy mostly comes from an increased number of True Positives, i.e., more cracks detected. In addition to improved hit rates, the processing speed for crack detection has been improved by a factor of four.

7. Corrosion detection

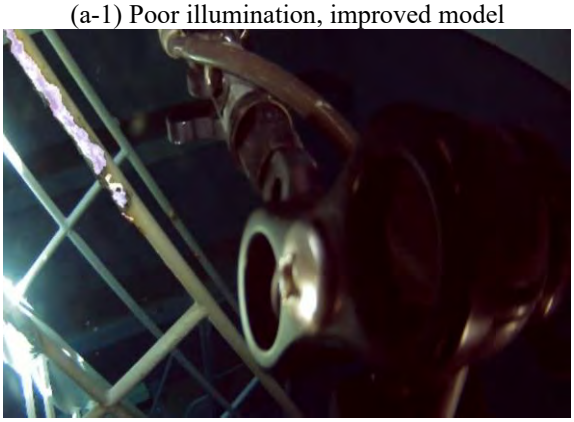
Corrosion detection performance is improved since *Stensrud et al. (2023).* The previous model had more false negatives in images with poor illumination or dark objects. Corrosion is often detected and assessed in the overall examination, unlike (small) cracks that are mostly detected in close-up examinations. Overall examination is performed at a larger distance from the surface, and the issue is that the lighting mounted on drones is often too weak to sufficiently illuminate the scene captured by the camera. The previous model also had more false positives when encountering objects or patterns that are not present in the training dataset, i.e. scenes that are out-of-distribution (OOD). Furthermore, its performance was strengthened against visual obscurity such as blur noise caused by image degradation, improving the accuracy of corrosion identification.

To reduce false positives under low-light conditions, the training data was augmented by synthetically generating dark patterns on images. To reduce false positives caused by out-of-distribution (OOD) image objects and to better handle novel patterns, we added images from the COCO dataset into the training set, used the model to detect likely corrosion, and selected false positives (images falsely

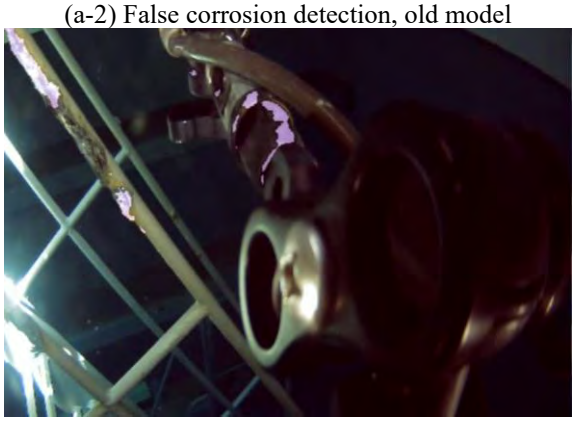
predicted as corroded) as negative samples for training. To address visual obscurity, we introduced blur and noise augmentations during training. The detection results of new vs. old model is illustrated in Fig.16. Fig.16a shows a frame captured from a drone-based inspection video. Due to the considerable distance from the wall and poor lighting conditions, the scene appears significantly underexposed. The previous model mistakenly identified corrosion in this area, but the updated model successfully eliminates this false detection. Fig.16b shows an image captured during an ROV underwater inspection. Visible corrosion is present on the structural surface in the upper-left corner, while a camera handle occupies the central area. The previous model mistakenly flagged parts of the dark camera handle as corrosion and failed to detect some actual corroded regions. The refined model corrects these issues by suppressing false positives and enhancing the identification of true corrosion.

(a-1) Poor illumination, improved model

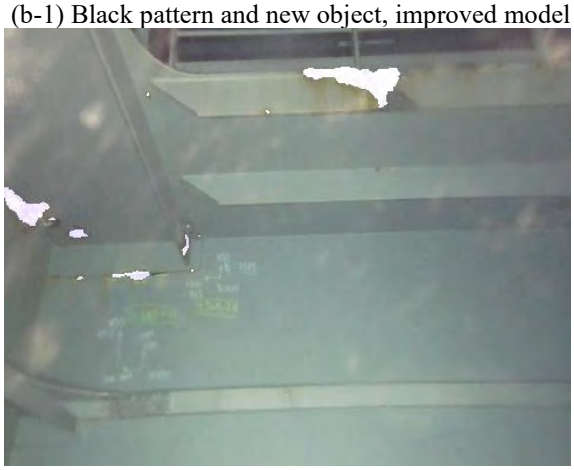
(a-2) False corrosion detection, old model



(b-1) Black pattern and new object, improved model



(b-2) False and incorrect detection, old model



(c-1) Visual obscurity, improved model

(c-2) Incorrect corrosion detection, old model

Fig.16: Improved corrosion detection example (The color differences of the violet corrosion pixels are due to the transparency layer, blending the violet color)

Fig.16c also displays an image captured during an ROV inspection. Visual obscurity is primarily due to underwater turbidity. The previous model failed to detect some corrosion in this scene, whereas the enhanced model successfully identifies the corroded areas.

A quantitative comparison between the old model and the improved model is provided in *Gong et al. (2025)*. That paper presents a scalable framework for deploying AI in remote ship inspections, using Corrosion.ai, *Wei and Chen (2021),* to show how continuous monitoring, data intelligence, and model adaptation can mitigate performance degradation. Some initial results, along with the uncertainty assessment in defect corrosion detection, are also reported in *Stensrud et al. (2025).*

8. Conclusions and further work

An asset-centric 3D digital twin has been developed and is continuing to be developed. 3D visualisations of geometric digital twins arguably improve the monitoring of the asset condition, but for most ships only 2D drawings exist, and existing systems like Ship Manager Hull require expensive, manual developments of a 3D model from 2D drawings. This research is investigating cheaper ways to create geometric 3D models from point clouds collected by drones and exploiting the features of parametric models. Currently, however, we are able to exploit the repetitive patterns of ballast water tank compartments to speed up the inspection using semantic technology and predictive path planning for the autonomous drone. The 3D digital twin accommodates several data types such as point clouds, videos, and location-tagged findings.

Automated detection of geometric defects in point clouds has been demonstrated. Pitting is also detected in point clouds, but not yet automatically. Automated crack and corrosion detection is steadily improving. There are fewer false positives and negatives than two years ago.

Image quality metrics have been investigated for the special case of close-up images of cracks. The current results are disappointing, but not unexpected in afterthought. Further work will be to investigate deep learning models and blind image quality assessment methods on this data. Another, more fundamental stream of research is to review the definitions of what a crack is. The existing definitions are targeted towards human experts, inspectors and surveyors with degrees in naval architecture and general high knowledge, competence, and human intelligence.

Acknowledgements

This research is partly funded by the Research Council of Norway through the AIDiT project, grant number 346524. Chaudhry Rehan Ikram, DNV, provided figures Fig.4, Fig.5, Fig.6, and Fig.7 and information about the status on the digital twin application development.

References

AHARON, N.; ORFAIG, R.; BOBROVSKY, B.Z. (2022), *BoT-SORT: Robust associations multi-pedestrian tracking*. arXiv preprint arXiv:2206.14651.

DHARMADHIKARI, M.; ALEXIS, K. (2025), *Semantics-aware Predictive Inspection Path Planning*, IEEE Trans. Field Robotics Vol.2

GONG, X.; ZHANG, H.; STEIRO, S. (2025), *Trustworthy and Resilient AI: A Case Study in Remote Ship Inspection*, 5th Int. Conf. Artificial Intelligence, Automation and Algorithms, Singapore

LI, C.; LI, L.; JIANG, H.; WENG, K.; GENG, Y.; LI, L.; KE, Z.; LI, Q.; CHENG, M.; NIE, W.; LI, Y. (2022), *YOLOv6: A single-stage object detection framework for industrial applications,* arXiv: 2209.02976

POGGI, L.; GAGGERO, T.; GAIOTTI M.; RAVINA E.; RIZZO, C.M. (2020), *Recent developments*

in Remote Inspections of Ship Structures, Int. J. Naval Architecture and Ocean Eng. 12, pp.881-891

STENSRUD, E.; GONG; X.L.; HAMRE; G.; LIU, Y.E.; LØVOLL, G.; McKAY, D.; RAEISSI, B.; RICHARDSEN, T.; SKRAMSTAD, T.; TORSTENSEN, A. (2025), *Artificial Intelligence as a Means to Automate Detection of Defects in Inspection of Hull Structures*, Offshore Technology Conf., Houston

STENSRUD, E.; BJØRDALSBAKKE, H. (2023), *Concept of a Digital New Hull Survey Process*, 22nd COMPIT Conf., Drübeck

STENSRUD, E.; KLAUSEN, K. (2022), *Another Step towards Remote Inspections of Maritime Vessels using Tailored Inspection Drones Instrumented with Computer Vision*, 21st COMPIT Conf., Pontignano

STENSRUD, E.; SKRAMSTAD, T.; BASHIR, M.; GARETT, J.; HAMRE, G.; KLAUSEN, K.; RAEISSI, B.; ROSSVOLL, P.; XIE, J.; ØDEGÅRDSTUEN, A. (2020), *Towards Remote Inspections of Maritime Vessels Using Drones Instrumented with Computer Vision and Hyperspectral Imaging*, 19th COMPIT Conf., Pontignano

STENSRUD, E.; SKRAMSTAD, T.; CABOS, C.; HAMRE, G.; KLAUSEN, K.; RAEISSI, B. (2019), *Automating inspections of cargo and ballast tanks using drones*, 18th COMPIT Conf., Tullamore

WANG, M. (2023), *Blind Image Quality Assessment: A Brief Survey*, https://arxiv.org/abs/2312.16551

WEI, Q.; CHEN, Y. (2021), *An AI-powered Corrosion Detection Solution for Maritime Inspection Activities,* 20th COMPIT Conf., Mülheim

ZHANG, Y.; SUN, P.; JIANG, Y.; YU, D.; WENG, F.; YUAN, Z.; WANG, X. (2022), *Bytetrack: Multi-object tracking by associating every detection box*, Eur. Conference Computer Vision (ECCV), pp.1-21

ZHANG, W.; ZHAI, G.; WEI, Y.; YANG, X.; MA, K. (2023), *Blind image quality assessment via vision-language correspondence: A multitask learning perspective*, IEEE/CVF Conf. Computer Vision and Pattern Recognition, Vancouver, pp.14071-14081

Integrating Revit into Shipbuilding Design

Joanna Sieranski, PROSTEP AG, Hamburg/Germany, joanna.sieranski@prostep.com

Abstract

An increasing number of European shipyards use Autodesk Revit as a design tool. This BIM tool supports planning and designing buildings but proved also to be useful in certain ship building scenarios. However, Revit has limitations as it neither provides shipbuilding-specific capabilities nor is it integrated with shipbuilding tools. Thus, the wish to use Revit in combination with other tools arises. As always, such constellations of different products introduce the need for automatic data transfer and synchronization. This paper illustrates the possibilities and practical implementation of such integration of the Revit software in shipbuilding.

1. Introduction

Building Information Modelling (BIM) is the answer to the digitalization and its opportunities in the field of architecture, engineering, and construction (AEC). It is a methodology to support the collaboration and understanding among all participants of a project. It spans the full lifecycle of the building project starting with the planning, continuing during execution, usage of the facilities, and ending with the exploitation. To achieve its goal BIM emphasises the usage of a central virtual model of the building and other needed data. The BIM software covers a variety of functions beyond the 2D and 3D representation like e.g. simulation and collaboration.

The utilization of the virtual representation in BIM has similarities to the concept of the Digital Twin (DT). DT also aims on creating and maintaining a virtual representation of a physical object like e.g. the whole ship. Both concepts claim to cover the whole life cycle of the products. However, the focus of these two concepts differs slightly. DT underlines the importance of operation and real-time exchange of data. BIM on the other had side concentrates on the collaboration on a construction project. It is more on the creation side rather than monitoring as it works with manual updates and thus a static data. Nevertheless, this focus, and maturity of BIM methodology and software proved to be useful in its field. Some governments like e.g. Norway require BIM when concluding public contracts.

Meanwhile also the ship building industry started to look at the tools and methodology used in BIM. For example, *Jensen (2018)* and *Preece (2019)* see the similarities between the on-land construction and design of cruise ships. Modern cruise ships often are not limited to the board and logging areas but contain additional entertainment spaces like theatres and swimming pools. The more complex these parts of the ship get the more they get into the focus of the design. Here is where the BIM knowledge and tools come in handy. At the same time, the concept of a central model for collaboration and the connection of 2D and 3D models is a universal idea that reduces errors, costs, and improves the overall quality.

At the same time ship building is a complex field with many differences to the land construction. These start already with how the steel structures are utilized and constructed compared to something like a building. There are also fundamentally different regulations and simulations needed for a ship. Things like watertightness are crucial for a ship but less so in the land AEC industry. The performance measures like hydrodynamics and structural integrity for marine vessels were not in focus of the original BIM method.

2. BIM Tools in Ship Building

The usage of a central model suggests that the best-of-suit approach, where all the software comes from the same vendor, would be the best way to adapt BIM in ship building. While this has benefits

Kannengießer (2023) reminds the limitations and additional efforts needed for successful implementation. An alternative is the best-of-breed tool selection that allows to select the best tool set regarding a certain task. This requires initial effort to integrate the tools but offers more flexibility for the shipyard.

As the application of BIM in ship building does not have to be limited to the processes and concepts only, shipyards may consider using BIM software as part of their best-of-breed approach. This allows to fully take advantage of the BIM idea and the maturity of the available tools in specific fields like e.g. interior design.

A popular tool among the BIM tools is Autodesk Revit. Its purpose is the planning, modelling, and documentation for construction projects. Revit supports multiple views at the same time including 2D and 3D representation with changes from one view being instantly reflected in the others. The basic building concepts like walls, mechanical and electrical parts are part of the software models. The objects are organized in so called families which might also be seen as catalogues depending on the context. The Revit model is stored in a central file with access granted to the different participants to support collaboration. *6sense (2025)* estimates Revit market share around 35% in its field and identifies it as one of the leading solutions. Thus, it is a good fit for yards considering the complementary usage of BIM software.

2. Integration

The previous chapters discussed that BIM might be a valuable addition to the ship building process. This chapter shows a practical application of Revit as a tool which might occur in a best-of-breed software environment. As BIM benefits at most in the planning and design phase this is the one examined here.

2.1. Use Case and Requirements

The first step in an integration approach is to evaluate and select the best tools for the specific tasks. As this paper concentrates on the application of a BIM tool based on Revit the question to answer is when it makes sense to use the software. Due to the lack of ship performance measures, it is better not to use Revit in the initial design. Instead, a good enough model could be re-created in Revit later for the outfitting. This is the part previously identified as like the on-land construction where the general layout comes together with equipment, wires, pipes etc. The tool supports pure geometry but more importantly also the domain specific data for the elements. The latter is the reason to use it in the first place. In addition, one could also use Revit to create renderings for presentation purposes, but this is not in scope of the discussion.

The example assumes that the main tool for the structural design and naval analysis is NAPA Designer. This is the source of information for the outfitting steps performed in Revit as target tool. Out of the box the only transfer possibility is the usage of a purely geometrical export and import. This may help for visualization but for working with the model the data would still be recreated using the native elements. To minimize the manual double work in the target tool, as much data as possible should be transferred automatically using the Application Programming Interfaces (API) provided in the source and target tool, Fig.1.

This leads to the question what information do we need in the target tool exactly. Many systems including NAPA and Revit allow customization of attributes and filling them with values. It might be desired to transfer such data as is to provide the information in the target tool. On the other hand, unnecessary data would clutter the user interface while it may be ignored. This applies also to standard attributes that are native to the systems. Note that some of them may be calculated like e.g. the weight of an element so it needs careful consideration what should be transferred. For the purposes of this paper the selection of concrete attributes will be ignored. The focus is on the general idea of transporting the information to the target system.

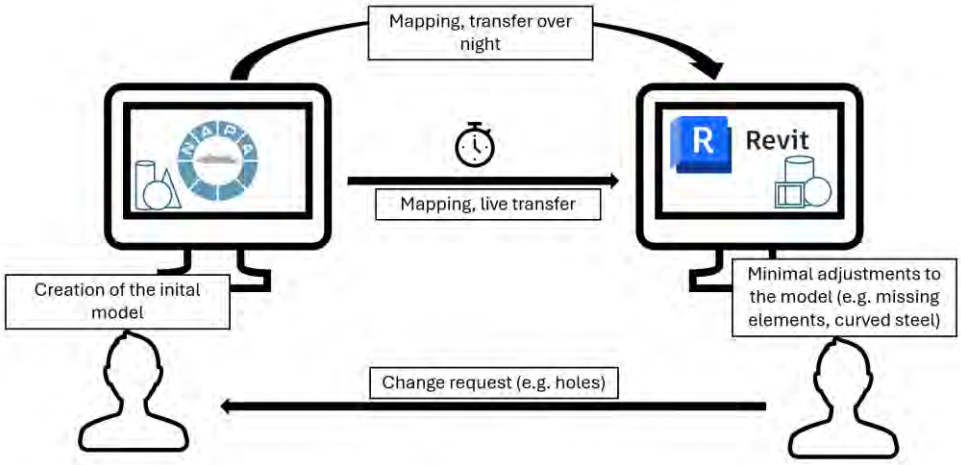


Fig.1: Transfer requirements and process outline

The most important attributes unveil when looking directly at the elements to transfer. For the outfitting, the volume and steel data designed in the source system play a crucial role. Thus, the transfer should re-create the panels with their stiffeners in the target system. The 3D data in Revit should have the correct thickness coming from NAPA so the designer can work with the correct space information. If the material exists in Revit this should also be assigned to the target object. Another aspect important for the work with the transferred model are the topological references like the limits of panels and volumes. Regarding the panel names, these could be transferred as is or changed to fit their purpose in Revit better. This could be done simply with a name mapping.

Keeping some kind of naming references is helpful for updates after an initial transfer. This is important as the ship design process consists of multiple iterations including feedback loops and changes. To keep order in this process it is good practice to have a leading system. The changes are then made in that system and propagated to the dependant system. Although Revit may play an important role in the design the leading system should obviously be a dedicated ship building tool. Changing the whole structure of the ship without proper support may be a dangerous thing. Thus, such changes should be communicated to the specialists in the source system and made there after checking the impact. The next step is then to update the data in Revit.

For setting up an automatic process one would also take the transfer times and frequency into account. In many cases an overnight transfer is sufficient, so the designers see the changes from the previous day when they come back the next morning. For simplicity no more detailed requirement regarding the runtimes is specified here. A transfer could also take place on-demand, but then additional constraints must be checked e.g. regarding changes on the model during the transfer.

2.2. Mapping

Although many of the concepts in the outfitting match between BIM and the ship building design these two are two different contexts. This is reflected in the tools which require translation of the ideas from ship building into the world of the BIM tool like Revit. This already starts with fundamental things like the panel representation. This chapter presents some of the mapping possibilities as a starting point for integration of the two tools. It is not meant as an exhaustive specification.

A panel could be easily seen as a wall or floor in a ship section. This is a good approximation, but it opens the question of how to determine if a panel should be a wall or a floor. The mapping could be a purely geometrical one where every horizontal panel is translated to a floor and perpendicular panels are the walls. For slanted panels the angle to one or the other could determine how to classify the object. For more control over the process the user could make the decision with a dedicated function attribute. This way everything set as deck could be mapped to a floor, or some elements could be

easily excluded from the transfer. Additionally, the interior and exterior side of the wall element could be used similarly to the material side of the ship steel.

Regarding the topological references, Revit supports that in different forms. The first one is the level representing an elevation. It is an important concept as most of the elements require a reference to a base level as shown in the Revit property “Base Constraint” in the right part of Fig.2. At first this might seem to be a big restriction resulting in a new level each time a panel, or a wall, doesn’t start on the deck. However, to cover such cases Revit allows specifying offsets from the base level. It provides more freedom for the placement of the elements like panels, but it also means that possible 3D coordinates from the ship system need to be defined relatively to the selected base level. Apart from that it makes it impossible for an element to ignore the base level e.g. at Z=0, and we always need to keep a topological reference there.

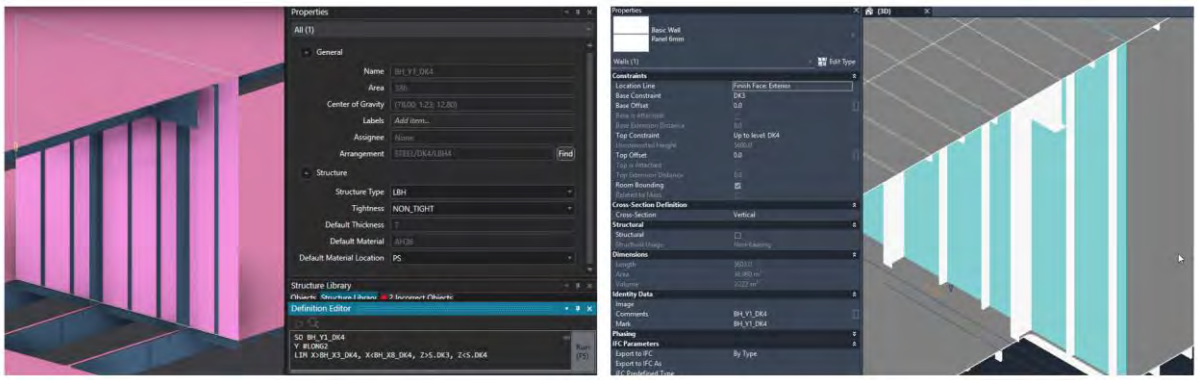


Fig.2: Standard panel properties in NAPA (left) and Revit (right)

The second possibility for references are joins between walls. If two walls are connected that way moving one of them will automatically extend the other. Joining walls is only allowed in certain places like e.g. when one wall or panel ends on the other. These connections are managed by the user with a checkbox to allow/disallow joins instead of typing some kind of id or name. Usually, Revit will find the connection automatically if “allow joins” is enabled and a join is permitted in a certain place. However, it is also possible to break a specific connection later. Another way to connect elements is the constraint locking mechanism in combination with architectural lines and alignment options with their own restrictions. For example, if there are two panels “on top” of each other the upper panel could be aligned to the molded plane of the bottom panel. Then, to make sure that the panels are always moved together a lock object is needed.

The stiffeners and flanges may be modelled in different ways, however some of them come with more restrictions than the others. If the stiffener should be attached to a non-horizontal panel automatically, a profile sweep could be the first try. These are always hosted by a wall with the benefit that inclining the wall or panel would also apply to the stiffeners. However, wall sweeps could only be horizontal or vertical with the tested Revit version and there are no sweep profiles on the floors. This limits the usage of the sweep profiles for stiffeners.

As we need to categorize a panel as a wall or floor (and a ramp for slanted decks) a uniform handling of the stiffeners may be a challenge. If we want to have the stiffeners align with the panel changes a separate Revit family with a host object offers more flexibility than wall sweeps. Such families are a collection of classified objects with common properties. However, even here we must either ensure that the catalogue works on panels and floors or create two catalogues, one where the host object is a horizontal deck and one where the host objects are slanted panels or walls. In both cases the stiffeners can be designed with variable or fixed parameters like e.g. a width of a web. This allows straight stiffeners that are not aligned to the horizontal or vertical main axis.

Another possibility to map stiffeners in Revit is the usage of vertical columns and horizontal beams.

In fact, there are catalogues for Revit that provide steel structures like the types used in shipbuilding e.g. the T or L-profiles. The concrete values may need to be adjusted, or new profile types modelled manually. When using columns and beams an important thing to know is that these are driven by the level and grid lines. While slanted columns and beams are allowed and may match the panel geometry at creation, they are always connected to a level and do not reference a floor or a wall directly. One can use locks to wire these two together as mentioned earlier but this may come with limited usage. Depending on how the alignment and lock were created changes to the wall may break the connection.

Modelling holes is an easy task in Revit if one chooses the correct Revit objects like a floor, wall or structural column. In such a case the holes are created on the proper face of the object and can be adjusted later e.g. moved to another location. As usual in Revit, if one wants to have parametric holes a family object is needed. Otherwise, the hole might be created freely by sketching it on a face.

The next step after the steel structures are in place is the volume transfer. For a ship this means the compartments enclosed by the steel structures or other definitions. In Revit volumes are represented by room and space objects. The room allocation requires a base level and usually walls that limit it. Alternatively, one may also use other architectural elements or architectural lines which are virtual objects only without a physical counterpart. Their purpose is to define architectural areas or in our case ship areas. The spaces have a different purpose as they are important for the mechanical and HVAC elements and allow to run some analyses. The spaces share the physical area with the rooms and may hold a reference to the room they are contained within.

After the mapping of ship design elements to BIM concepts a more detailed look raises the question regarding the transfer of the attributes. For example, a panel mapped to a wall “loses” some of the attributes or rather they may be stored differently. While the thickness of a panel might be an attribute on a single plate in a ship design tool in Revit the data is hold by the wall type which is here combined with the material information.

Table I: Mapping examples

Source (Ship)	**Target (BIM)**	**Remarks**
Perpendicular panel	Wall	
Deck panels	Levels and Floors	Levels are mandatory to place other objects like walls or columns
Panel name	Wall mark or custom attribute	Allows duplicates in Revit
Material by name	Material by name	Material must be added to catalogue first
Panel thickness	Wall type combining thickness and material	Wall types must be added to catalogue first.
Panel material		
Panels with mapping issues	Paint on a wall	Paints may use different material and are independent from the wall type, no effect on thickness.
Stiffener type	Structural stiffener type	Must already exist in project. Type can contain variable parameters (e.g. web height and width) or different dimension sets are mapped to its own type (e.g. 80x4, 120x6). Stiffener direction determines flange side.
Stiffener offset	Build in offset field	
Room description	Room comment	
Room volume	Room/Space volume	Computed automatically by the systems
Tightness	Custom parameter	Informative purpose only

At the same time other data might not be set directly either because it is calculated by the software (e.g. a volume of a room) or because there exists no matching parameter (e.g. tightness). For the latter case it is possible to define custom parameters and set them on the desired objects.

An extensive attribute mapping would be too much for this paper, but some examples can be found in Table I. Note that for updating the target model in Revit in an automated way an id or name mapping is crucial. For a human user it might be intuitive to find the elements and change them. For a machine, the usage of a name mapping is usually the most secure and efficient way. In the best case we can use the identifying fields from the source and target systems e.g. deck panel name in NAPA and a level name in Revit. However, some elements in Revit do not provide a matching property. For example, a room here has attributes like room number to easier distinguish it from other rooms but apart from a warning nothing hinders the user to enter the same number in two different rooms. For the sake of better automation and recognition of rooms in later updates, customization might be needed.

2.3. Automating the Integration

In the previous chapters the use case and basic requirements were described. This chapter concentrates on an example implementation for automatic integration. The source tool with the steel design is NAPA Designer. The NAPA software is a conceivable source of information which could be used in a real scenario for Revit integration. Additionally, an export plugin for the steel structures already exists. Though its purpose is the transfer of the data to AVEVA Marine, it contains the needed panel and stiffener information. This might be re-used in an import into Revit. What the current implementation lacks is the export of any information regarding the rooms. Using the NAPA C# API it would be possible to change this, but the first transfer attempt concentrates on the utilization of already available data. Note that the native data and APIs in NAPA Designer and Revit are used, rather than a neutral 3D format. The reason is that the latter often is supported in a limited way on both export and import side. For example, the data is visualized with proper geometry but doesn't provide access to parameters and types. Using the APIs directly in the tools provides us with important information and possibilities to re-create the elements on the target side in a more convenient way with preservation of additional information.

Revit provides its own programming interface to enable third parties the implementation of plugins for the software. PROSTEP uses this currently to develop a connector for the import, export, and update of the data in Revit. The connector software is designed to be used generally for different scenarios related to the integration with Revit software. Thus, it makes sense to use it for the integration of ship building data. Due to the nature of such data, an additional basic mapping method for ship information is implemented in the connector. It operates in the same format as the data coming from NAPA export. Between the export from NAPA and the import into Revit the file could also be modified using a variable mapping. This makes the process more flexible as it doesn't require changes to the code in the tools. At the same time the mapping can be automated with a toolset that can handle the file format.

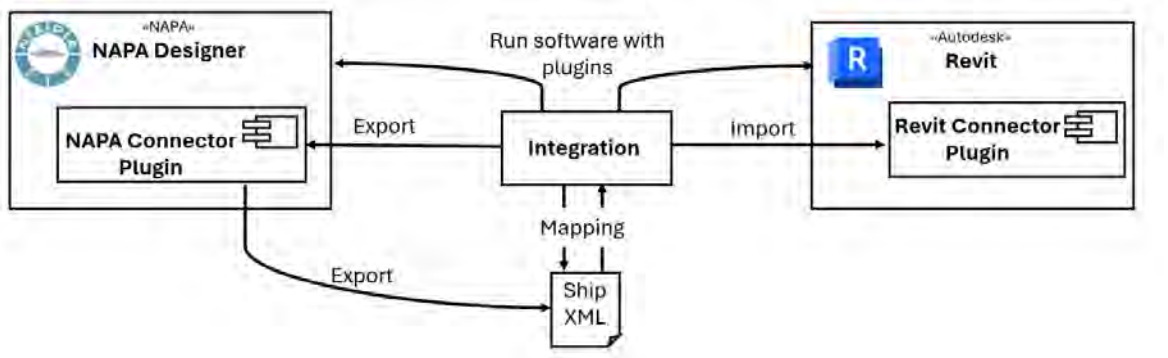


Fig.3: Example integration process

While technically the solution on the Revit side consists of a Revit plugin it isn't meant to be used manually by a human. Instead, the plugin is designed to be called by a wrapper application which handles the integration process automatically as outlined in Fig.3. Ideally the same would happen on the export side, after adjustments in the code there. For now, the file is exported manually using an existing NAPA plugin, then modified by a script and passed to the Revit connector using a HTTP REST interface.

While the source and the target tool provide their APIs based on a plugin idea their support for automatization differs from each other. Both tools have in common that the source and target software must be started to enable working with the data. However, NAPA offers a headless mode where the application can be started in the background. Revit does not provide this possibility, so we always have an open Revit window during the integration. Additionally, modifications of the opened files may lead to pop-ups that usually would be answered by the user. An integration solution must take care of this. As there is no extensive list of all the possibilities, this often leads to a trial-and-error approach where a new warning or question needs to be handled with a new case. The fallback here is always to stop the integration if something unexpected occurs. Finally, the work with an open window in Revit goes in hand with a Revit-driven event mechanism. This means that while the integration may trigger or "request" actions, the Revit software decides when these are carried out. With some adjustments in the integration code, good enough results regarding response times could be achieved. Luckily on the export side the integration code can be run directly without any interim layers or additional context requests from the NAPA side.

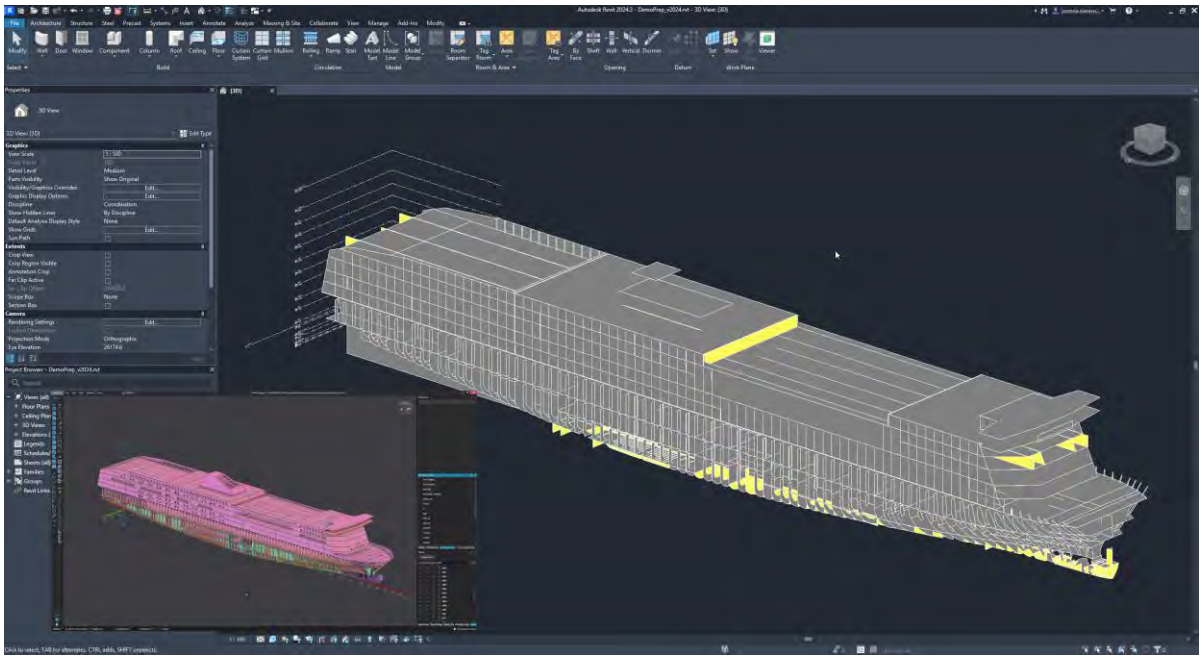


Fig.4: Planar panel transfer from NAPA Designer to Autodesk Revit

In the end our basic integration implementation managed to transfer planar panel geometry including stiffeners between NAPA Designer and Revit. Note that the examples in Figs.4 and 5 are work in progress and their purpose is the evaluation of what can be achieved. Thus, some steps were run manually. This includes starting the export with a set of parameters and running the HTTP REST call with the requested file. These steps are easy to automate but time-consuming and not important when we look at the data itself. What is important here is that not only the geometry of the object but also a portion of the parameters and references came through in a meaningful way.

The integration handled the planar panels and mapped them to floors in case of a deck and to walls otherwise. In the first attempt, only perpendicular walls were mapped, thus the slanted walls are missing in Fig.4. The panels retain their exact shape but currently no openings were transferred.

However, with the already implemented code for geometry mapping in panels and parameter mapping in stiffeners the step to a proper hole mapping is clearly doable. The panels mapped to walls keep their references to the decks above and below by using the base level and top-level fields in Revit. These references are determined in two ways: by checking the limit information obtained from NAPA and by geometry if no limit could be found. The latter is needed as a wall in Revit requires a reference to a base level. If no direct matching level can be found, the next one is selected and an offset places the panel in the correct position.

Other limit information, like limit by a neighbor panel, is not set currently. Instead, the Revit joining mechanism mentioned earlier is used e.g. panels ending at each other are connected automatically if possible. The next step could be checking these connections against the limit information from the panel and removing them if needed. Additionally, an investigation of the constraint locking mechanism would be beneficial to check if the integration could utilize that with the already existing limit information. This could also be used in combination with other parts.

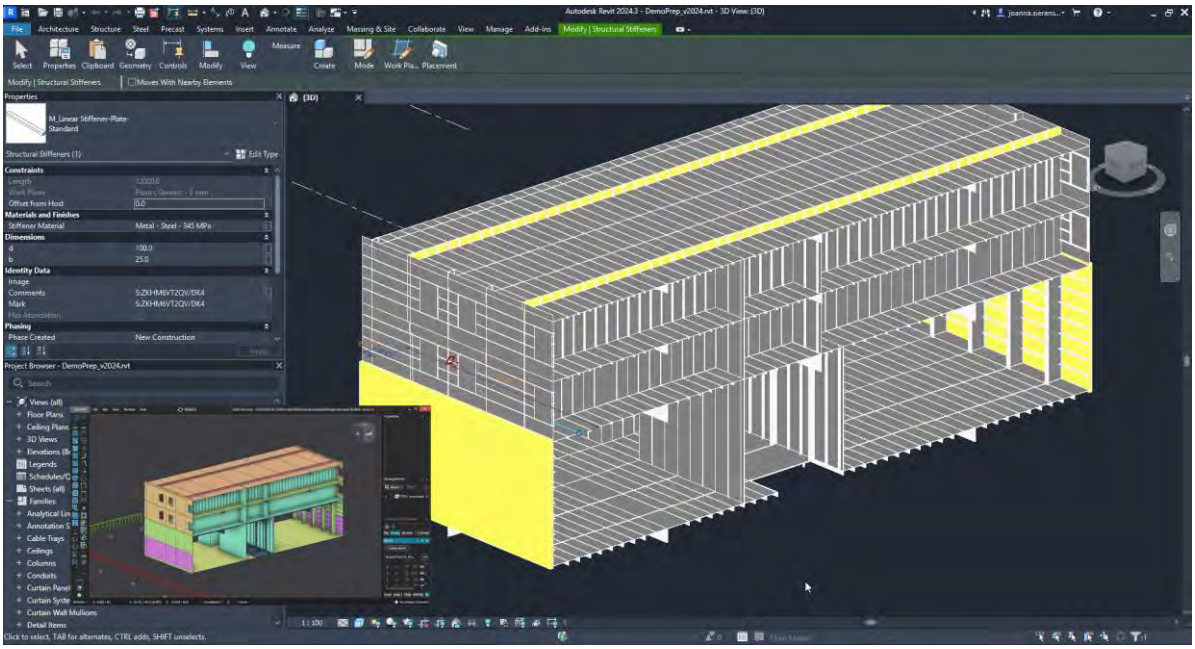


Fig.5: Block transfer including stiffeners

The stiffeners in Fig.5 were modeled as a structural stiffener family in Revit which means that they automatically know their parent panel and will be moved accordingly. With a mapping to the proper Revit family, they keep their shape and a set of parameters like the width of a web. Of course, this depends on the family used in Revit, the better the target model the more exact the mapping can be. Same is true for the panel thickness: if there is no matching wall type in Revit the panel must either be ignored or a default wall type with default thickness must be selected. In the past transferring elements with default values and coloring them turned out valuable. In Revit this can be achieved by the paint feature or be setting a different material depending on what the element supports. For illustration purposes some of the panels in Figs.4 and 5 weren't mapped with the incorrect wall type and had wrong thickness. Their color was adjusted accordingly during the transfer.

3. Conclusion and Outlook

The previous chapters showed that there are enough similarities between ship building and BIM that the tools for the latter can be used in the case of outfitting. This is most beneficial in a field like cruise vessels. A first implementation of a semi-automatic approach by PROSTEP checked if a combination of the tools from both worlds may be supported to reduce manual work. The efforts were concentrated on the most basic parts of the ship needed in outfitting: planar panels and their stiffeners. These could be transferred to wall, decks, and stiffeners with proper setup which reduces the manual work.

Different colouring of the walls supports the users to find elements that need to be re-checked if errors occur.

However, the transfer may still be improved. PROSTEP already started to check the possibility of room data transfer between NAPA Designer and Revit as well as from Revit to another ship building tool that is Intergraph Smart Marine 3D. We are confident that a good transfer quality may be obtained here as well. Staying in the ship steel field further improvements are possible. Transferring of omitted elements like e.g. brackets or stiffener endcuts may be desirable. At the same time a look at automatic constraint creation and locking seems to be good to try, if further references in the already transferred model are needed. The presented thoughts and implementation regarding a mapping between ship design and BIM software is a good start to investigate these additional use cases. The automatic transfer has a lot of potential to save manual work and reduce errors.

References

JENSEN, M. (2018a), *BIM on the high seas?,* https://develop3d.com/opinion/bim-on-the-high-seas-fraiserline-ship-design-marine-engineering/, Develop3D

JENSEN, M. (2018), *BIM on the high seas?,* https://www.cruiseandferry.net/articles/can-building-information-modelling-work-for-cruise-ships, Cruise and Ferry

WIMMER, R.; BARTELS, N.; MAILE, T. (2025), *BIM Professional Next Generation BIM für Praxis und Lehre,* building SMART Deutschland e. V., bSD Verlag, Berlin, pp.23-34

GUALENI, P.; PETACCO, N.; ZINI, A. (2023), *The Role of the Digital Twin for a Ship Life Cycle Perspective: A Focus on Naval Vessels*, COMPIT Conf., Drübeck

KANNENGIESSER, R. (2023), *Enable the Best Shipbuilding Toolset*, COMPIT Conf., Drübeck

PREECE, E. (2019) *Virtual Reality and BIM: The future of shipbuilding?,* https://cruiseshipinteriors-europe.com/virtual-reality-and-bim-the-future-of-shipbuilding/, Cruise Ship Interiors

6SENSE (2025), *Market Share of Autodesk Revit*, https://6sense.com/tech/bim-and-architectural-design-software/autodesk-revit-market-share, 6sense

From Conventional to Bio-Inspired Robotic Platforms

Marco Bibuli, CNR-INM, Genoa/Italy, marco.bibuli@cnr.it
Angelo Odetti, CNR- INM, Genoa/Italy, angelo.odetti@cnr.it
Elena Ciappi, CNR-INM, Rome/Italy, elena.ciappi@cnr.it
Elena Paifelman, CNR-INM, Rome/Italy, elena.paifelman@cnr.it

Abstract

This paper focuses on innovative design and modeling approaches of biomimetic underwater platforms, corroborating such methodology by the development of an operational robotic fish, in the framework of the project PERSICO (Silent Robotic Fish for Observation and Sampling). The robotic fish prototype is designed for long-range, low-noise operations, integrating modular architecture, adaptive buoyancy control, and multi-jointed caudal and pectoral fins to achieve efficient propulsion, maneuverability, and reduced in-water noise. The paper details the vehicle's structural design, hydrostatic modeling, actuation systems, and dynamic behavior, supported by experimental trials in controlled and open-water environments. Mathematical models for surge, yaw, and heave dynamics are derived and validated, laying the groundwork for future fully autonomous operations and acoustic-based remote control. PERSICO represents a significant advancement in underwater robotics, offering a versatile and silent platform for dual-use applications in complex marine scenarios.

1. Introduction

The exploration and monitoring of marine environments have undergone a transformative evolution over the past two decades, driven by the rapid advancement of autonomous robotic technologies. From surface vessels to underwater vehicles, these systems have matured into reliable platforms capable of executing complex missions with minimal human intervention. Their integration into scientific, industrial, and environmental operations has enabled unprecedented access to remote and hazardous marine zones, facilitating high-resolution data acquisition for applications ranging from bathymetric mapping and ecological monitoring to infrastructure inspection and resource exploration.

Autonomous surface vehicles (ASVs), *Odetti et al. (2020),* and autonomous underwater vehicles (AUVs), *Odetti et al. (2017),* have become indispensable tools in oceanography and maritime operations. Equipped with advanced navigation systems, acoustic sensors, and modular payloads, these platforms offer scalable, cost-effective solutions for persistent and adaptive data gathering. The consolidation of these technologies has led to the development of standardized mission protocols, robust communication architectures, and interoperable software frameworks, marking a significant milestone in the operational maturity of marine robotics, *Bibuli et al. (2022).*

Yet, as the demand for more agile, efficient, and environmentally integrated systems grows, the field is witnessing a paradigm shift toward a new generation of bio-inspired robotic platforms, particularly robotic fish. These systems emulate the morphology and locomotion strategies of aquatic organisms, offering enhanced maneuverability, stealth, and energy efficiency in complex underwater environments, *Rus et al. (2015)*. Unlike traditional propeller-driven vehicles, bio-inspired robots can navigate through cluttered spaces, interact more naturally with marine life, and operate with reduced acoustic signatures, making them ideal for both ecological studies and covert operations.

The development of robotic fish introduces a host of scientific and technological challenges, *Junzhi et al. (2004).* These include the design of compliant materials and actuators that replicate biological motion, the integration of distributed sensing and control systems, and the modeling of fluid-structure interactions in dynamic aquatic environments. Moreover, achieving autonomy in such platforms requires novel approaches to perception, decision-making, and adaptive behavior, often inspired by the neural and behavioral mechanisms of real fish.

Beyond their scientific intrigue, bio-inspired robotic fish hold significant promise for dual-use applications. In the civil domain, they can be deployed for environmental monitoring, pollution tracking, and underwater infrastructure inspection in sensitive habitats. In the military sphere, their low observability and biomimetic movement patterns make them suitable for surveillance, reconnaissance, and mine detection missions in contested littoral zones.

This paper presents a comprehensive overview of the current state and future directions of autonomous marine robotics, with a particular focus on the transition from conventional platforms to bio-inspired systems. It explores the technological innovations, interdisciplinary challenges, and operational scenarios that define this emerging frontier, offering insights into how robotic fish may redefine the capabilities and roles of autonomous systems in the marine domain.

2. Concept of the Fish Motion

Fish-inspired biomimetic autonomous underwater vehicles (BAUVs) have garnered significant interest across research and industry due to their potential to replicate not only the propulsion mechanisms of aquatic animals but also their stealth, agility in constrained environments, and adaptability to unstructured conditions, *Sfakiotakis et al. (1999)*, *Li et al. (2022)*, *Zhang et al. (2024)*. Central to these designs is the selection of a locomotion strategy, which profoundly influences performance trade-offs in speed, endurance, stability, and payload capacity.

Locomotion modes in nature are broadly categorized into Body-Caudal Fin (BCF) and Median-Paired Fin (MPF) strategies, each defined by the extent and region of body undulations. In BCF locomotion, Fig.1, anguilliform swimmers like eels exhibit full-body undulations with high bending amplitudes, offering superior maneuverability in cluttered or narrow spaces but at the cost of hydrodynamic efficiency and speed. Subcarangiform and carangiform swimmers concentrate motion in the posterior body, balancing efficiency and mobility, ideal for mid-range missions. Thunniform swimmers, such as tuna, isolate propulsion to the caudal fin while maintaining a rigid anterior body, achieving high-speed cruising and excellent efficiency at high Reynolds numbers, making them suitable for long-range operations. Ostraciiform swimmers, with nearly rigid bodies and tail-only oscillations, prioritize stability and precise hovering, though they sacrifice speed.

MPF locomotion, observed in rajiform, balistiform, and labriform species, employs paired or median fins for propulsion. These modes enable fine control and station-keeping at low speeds, making them ideal for inspection tasks and operations near structures. Each locomotion type inherently dictates vehicle geometry and actuation complexity: thunniform designs favor streamlined, elongated bodies with minimal drag, while anguilliform and MPF systems require flexible structures and multi-fin actuation for enhanced maneuverability.

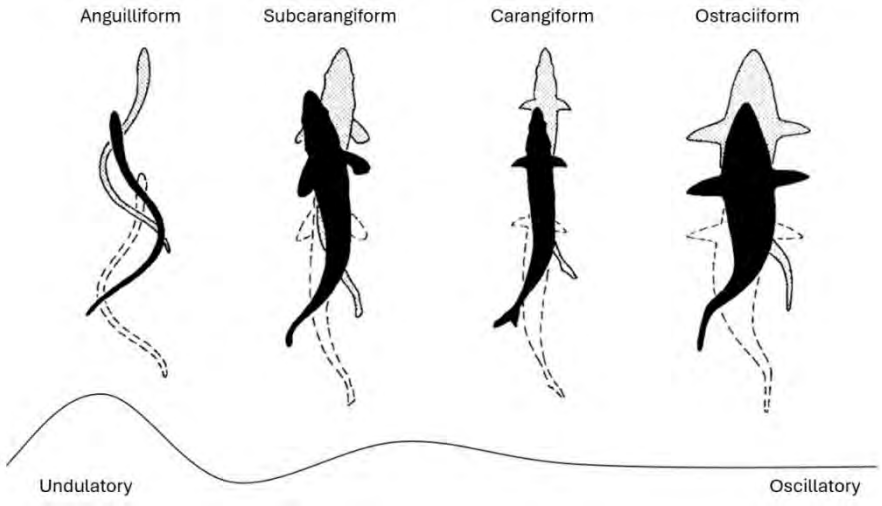


Fig.1: Body-Caudal Fin (BCF) motion classification

Despite the diversity of existing platforms, many BAUVs still face limitations. Few successfully integrate long-range endurance, flexible payload capacity, hybrid autonomous/tethered operation, and gliding capabilities. High-speed thunniform vehicles often compromise mission duration, while highly maneuverable anguilliform or MPF systems tend to have limited operational scope.

To address these trade-offs, this work introduces PERSICO (Italian acronym for Silent Robotic Fish for Observation and Sampling), a novel fish-like BAUV inspired by thunniform BCF locomotion. Designed for sustained cruising and long-range missions, PERSICO features a high-thrust caudal fin mounted via a modular interface, allowing rapid tail configuration changes based on mission needs. The vehicle incorporates adaptive swim bladders for buoyancy control and gliding, actuated pectoral fins for trim and maneuvering, and a compact yet spacious internal architecture to support advanced sensor payloads. Enhanced with onboard processing and redundant communication systems, PERSICO supports both autonomous navigation and tethered ROV operations. This integrated design consolidates the most effective features of contemporary fish-like BAUVs into a single, versatile platform optimized for endurance, efficiency, and maneuverability.

3. Design

The structural architecture of the PERSICO prototype is built around a modular framework composed of high-density polyethylene (HDPE), selected for its durability, buoyancy characteristics, and ease of machining. Inspired by tuna-like morphologies, the biomimetic body plan ensures hydrodynamic efficiency while offering internal flexibility for component integration. The internal layout houses pressure-resistant compartments for critical electronics and batteries, buoyant foam inserts for weight compensation, and provisions for a future outer fairing made from lightweight, hydrophobic materials. This configuration balances mechanical robustness with neutral buoyancy and streamlined submerged performance.

All modules were designed with an emphasis on modularity, reliability, and accessibility. In parallel with the structural development, a comprehensive hydrostatic model was constructed to ensure neutral buoyancy, static trim equilibrium, and passive underwater stability. This model was based on a detailed physical inventory of the vehicle's architecture, where each component was characterized by its dry weight, displaced volume (buoyancy), and spatial coordinates within the internal reference frame. These parameters were compiled into a computational spreadsheet to evaluate the overall hydrostatic response of the assembled system.

The model incorporated over 30 distinct elements, including structural covers, internal frames, propulsion units, electronics, cabling, and payloads. Each component was assigned a position vector, a weight in air, and a buoyancy value. Static moments of weight and volume were calculated to assess their influence on the vehicle's balance. The resulting vertical offset of approximately 49.3 mm between the center of buoyancy (COB) and center of gravity (COG) confirmed positive static stability in pitch and roll. Minor longitudinal ($\Delta x = -3.6$ mm) and lateral ($\Delta y = -1.1$ mm) offsets indicated a well-balanced design with no significant asymmetries.

Detailed analysis revealed that the forward-mounted payload, including the camera, CTD sensor, and DVL, significantly influenced pitch dynamics due to their mass being located over 1 meter forward along the X-axis. This was counterbalanced by the battery pack positioned near the aft section at $x = -280$ mm. A central foam block, providing 7.69 kg of lift, was strategically placed at the geometric center of the hull to support buoyancy. Additionally, swim bladder systems located at the bow and stern enabled trim control, active gliding, and buoyancy regulation. Even minor components such as cabling and metallic fasteners were tracked and positioned to minimize cumulative moment arms.

The design process followed an iterative workflow: initial component placement in CAD was followed by hydrostatic analysis to compute mass moments, total buoyancy, and static offsets. Refinements were made to bring the system within predefined thresholds, including net buoyancy deviations under 0.5 kg, a minimum COB–COG vertical separation of 30 mm, and centered balance across all axes.

The resulting final prototype platform has the following dimensions: 1.47 m (Length) x 0.35 m (Height) x 0.21m (Width) (the width is 0.59 m considering the lateral fin occupation), for an overall weight of 27 kg (in saltwater configuration).

The PERSICO vehicle features a biomimetic propulsion system centered on a multi-jointed caudal fin, designed to replicate thunniform locomotion while enabling advanced maneuvering capabilities. The fin incorporates two primary actuated joints: a basal hinge connecting it to the hull and an intermediate joint along the shaft, providing two degrees of freedom for controlled undulatory motion. A third actuator enables axial rotation of the entire fin, supporting roll stabilization and dynamic gliding behaviors. This configuration allows experimental tuning of fin dynamics and supports multiple actuation modes, including pure thunniform oscillation and hybrid configurations with flexible trailing segments. Fig.2 illustrates these modes, including an alternative whale-like rotation scheme for gliding and control surface functionality.

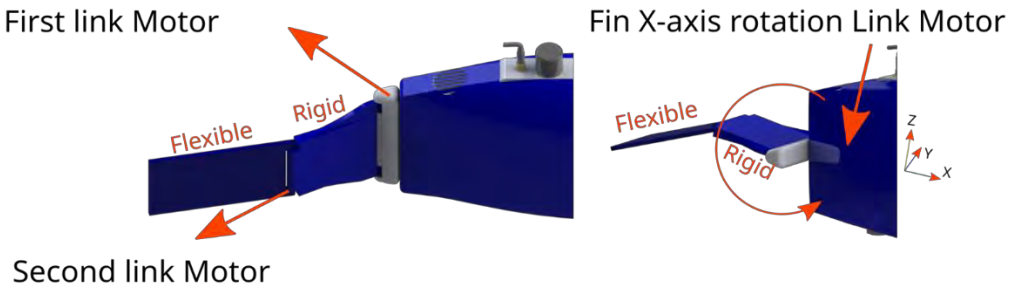


Fig.2: Caudal fin and joint actuation

Complementing the caudal fin, actuated pectoral fins contribute to pitch and yaw stability, particularly during low-speed operations and gliding transitions. Buoyancy modulation is achieved via mechanically actuated swim bladders located at the bow and stern, each driven by brushless motors coupled to worm gear assemblies for precise volume control.

The entire actuation system was engineered for efficiency, compactness, and modularity.

All propulsion units, control surfaces, and buoyancy systems are powered by miniature brushless DC motors selected for their high torque density, low acoustic signature, and integrated control electronics. These motors are housed in pressure-compensated capsules filled with dielectric oil, ensuring thermal dissipation and eliminating pressure gradients at operational depths. Dual radial shaft seals provide long-term mechanical isolation from the external environment.

The full actuation suite includes:

- Two motors for pitch oscillation of the caudal fin
- One motor for axial fin rotation (roll stabilization and gliding)
- Two motors for swim bladder control (fore and aft)
- Two actuators for pectoral fin trim and stability

Actuator specifications were derived from simulation-based performance envelopes, incorporating angular excursion requirements, dynamic torque profiles, and steady-state operating speeds. Manufacturer torque-speed curves were matched with resistive force models to determine optimal gearing strategies, including worm-drive configurations for high-load, low-backlash applications.

A critical subsystem is the mechanically actuated swim bladder, which modulates buoyancy via volume displacement. Each bladder consists of a sealed telescopic piston-cylinder chamber, driven by a motorized worm gear. The buoyant force generated follows Archimedes' principle:

$$F_B = \rho_{water} \cdot g \cdot \Delta V$$

Assuming seawater density ρ_{water}=1025 kg/m^3 and gravitational acceleration g=9.81 m/s^2, a target buoyant force of 3 N per bladder requires a volume change:

$$\Delta V = 1025 \cdot 9.813 \approx 298.6 \text{ cm}^3$$

This was conservatively rounded to 300 cm³ per unit to ensure control stability. The piston-cylinder system was dimensioned to achieve this displacement over a 100 mm stroke, resulting in a cross-sectional area of approximately 30 cm². Required actuation torque was calculated using a static equilibrium model of the worm-drive system, accounting for screw geometry, frictional losses, and mechanical backlash.

This integrated actuation architecture enables precise control, efficient propulsion, and adaptive buoyancy, supporting the PERSICO platform's mission versatility across a wide range of underwater scenarios.

4. Modelling

In this section, the mathematical model developed to evaluate the propulsive performance of the caudal fin of a fish exhibiting BCF (Body Caudal Fin) locomotion is presented. The fin consists of a rigid element connected to a flexible part, as shown in the prototype diagram in Figure 1. This design choice allows for the analysis of both carangiform and ostraciiform fish types, those characterized by fins made of both rigid and flexible parts, or by a single flexible element.

All model results refer to the mid-span section of the fin (highlighted in red in Fig.3), allowing the neglect of 3D edge effects.

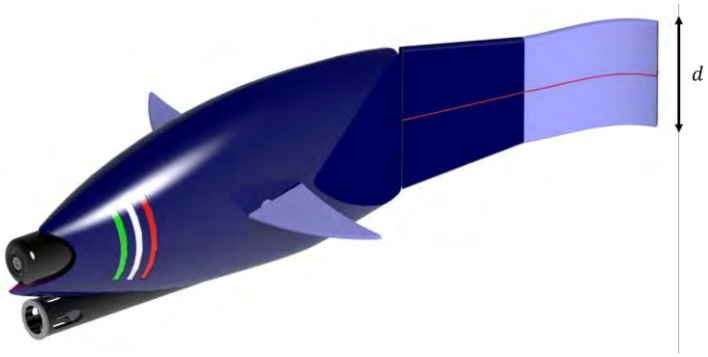


Fig.3: Concept 3D model of the robotic fish

4.1. Oscillatory Motion Modeling

To replicate the oscillatory motion of a fish, the system is driven by a rotation and a vertical displacement, both applied to a rigid section, as depicted in Fig.4. These motions are typically sinusoidal:

Rotation: $\alpha(t) = A_\alpha \sin(2\pi f_\alpha t)$
Vertical displacement: $h(t) = A_h \sin(2\pi f_h t)$

Where:
- A_α, A_h are amplitudes
- f_α, f_h are frequencies

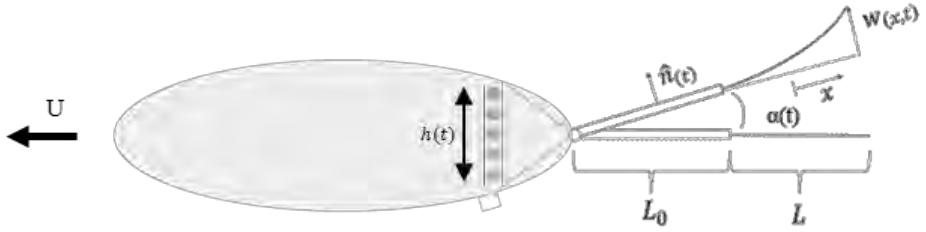


Fig.4: Top view of the fin motion modelling

4.2. Flexible Fin Model

The caudal fin is modeled as a Euler-Bernoulli beam, with its midpoint as the origin. The beam equation is:

$$EI \cdot w'(x,t) + \rho A_b\left[\ddot{\vec{u}}(x,t) \cdot \hat{n}(t)\right] = F_f(x,t)$$

E the Young's modulus, *I* the moment of inertia, ρ the fluid density, A_b the cross-sectional area, *w(x,t)* the beam deflection, $F_f(x,t)$ the hydrodynamic forces, *u(x,t)* the total displacement (rigid + flexible), *n(t)* the normal direction.

Boundary conditions: clamped at one end, free at the other end

$$w\left(-\frac{L}{2},t\right) = 0 \quad , \quad w'\left(-\frac{L}{2},t\right) = 0 \quad , \quad w''\left(-\frac{L}{2},t\right) = 0 \quad , \quad w'''\left(-\frac{L}{2},t\right) = 0$$

4.3. Hydrodynamic Model

Hydrodynamic forces are derived from pressure differences across the fin:

$$F_f(x,t) = -\Delta P(x,t)d$$

Pressure is split into non-circulatory due to added mass and circulatory due to vortex shedding.

Using Wu's potential flow theory, the pressure components are:

$$\Delta P_{NC}(x,t) = A_1(x)\ddot{w}(x,t) + A_2(x)\dot{w}'(x,t) + B(x)\dot{w}(x,t) + C(x)w'(x,t) + D(x,t)$$
$$\Delta P_C(x,t) = E(x)\dot{w}(x,t) + F(x)w'(x,t) + G(x,t) + L(x,0)\varphi(t) + g(x)\underline{E}(\ddot{w}(x,t) \cdot \varphi(t))$$

4.4. Coupled Fluid-Structure System

Using modal decomposition, the beam deflection is expressed as:

$$w(x,t) = \sum_{i=1}^{N} \Phi_i(x)q_i(t)$$

where $\varphi_i(x)$ are the mode shapes and $q_i(t)$ are the time-dependent modal coordinates.

The final coupled dynamic system becomes:

$$M\ddot{q}(t) + C\dot{q}(t) + Kq(t) + [\Psi\ddot{q}(t) + \Upsilon\dot{q}(t)] * \varphi(t) = F$$

where *M*, *C*, *K* are the mass, damping and stiffness matrices respectively, Ψ and Y are convolutionary

coefficients representing the generated forces of vorticity release, *F* are the external forces from rigid body motion.

The natural frequencies of the system are: $f_i = 2\pi\sqrt{\frac{K}{M}}$

5. From concept to reality

In the early days of the PERSICO project, the design concept of the robotic fish evolved from scratch ideas, through functional schemes and draft proposals, towards the imaginary final form of the prototype, as depicted in Fig.5 where a slender and agile robot platform is envisioned. Anyway, the reality has to match with the expectation, thus further refinements of the initial concept and design was carried out, keeping into account the dimension of commercial devices (actuators and devices), material and assembly constraints, payload, to name a few, leading to more feasible constructive design as shown in Fig.6, where the space and location of the different modules, control surfaces, frame shape, are taken into account. The prototype design review with the final dimensions in reported in Fig.7.

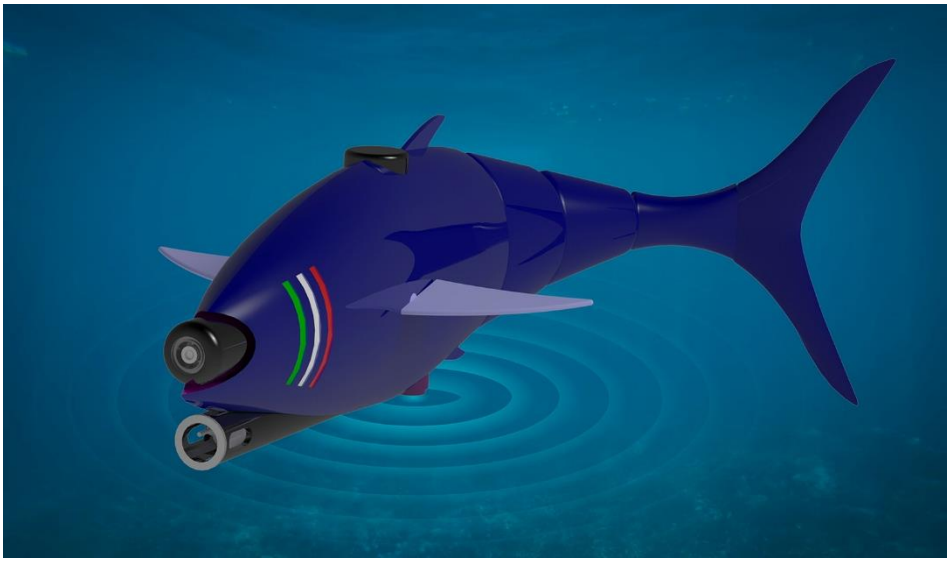


Fig.5: Conceptual representation of the robotic fish

The functional brain of the robotic fish is based on a commercial Raspberry Pi 3B board that has the ability to connect to the sensing devices for proprioceptive data gathering and to the actuation motors for controlling its motion, by means of the movable fins and internal bladders. A number of additional software modules are added to the control architecture in such a way to provide the autonomous functionalities related to the navigation, guidance and control, allowing the platform to perform advanced operations in the operational scenario. The robotic fish prototype was then realized, following the design specifics, resulting in the operational platform depicted in Fig.8.

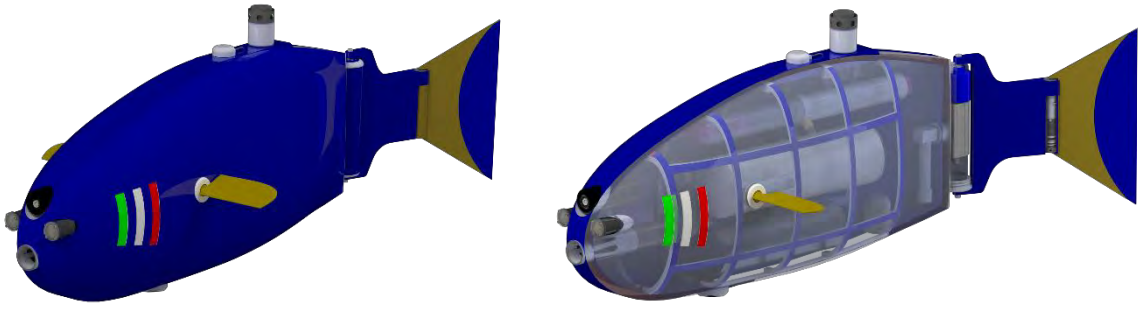


Fig.6: Constructive designs, outer shape (left), inner components allocation (right)

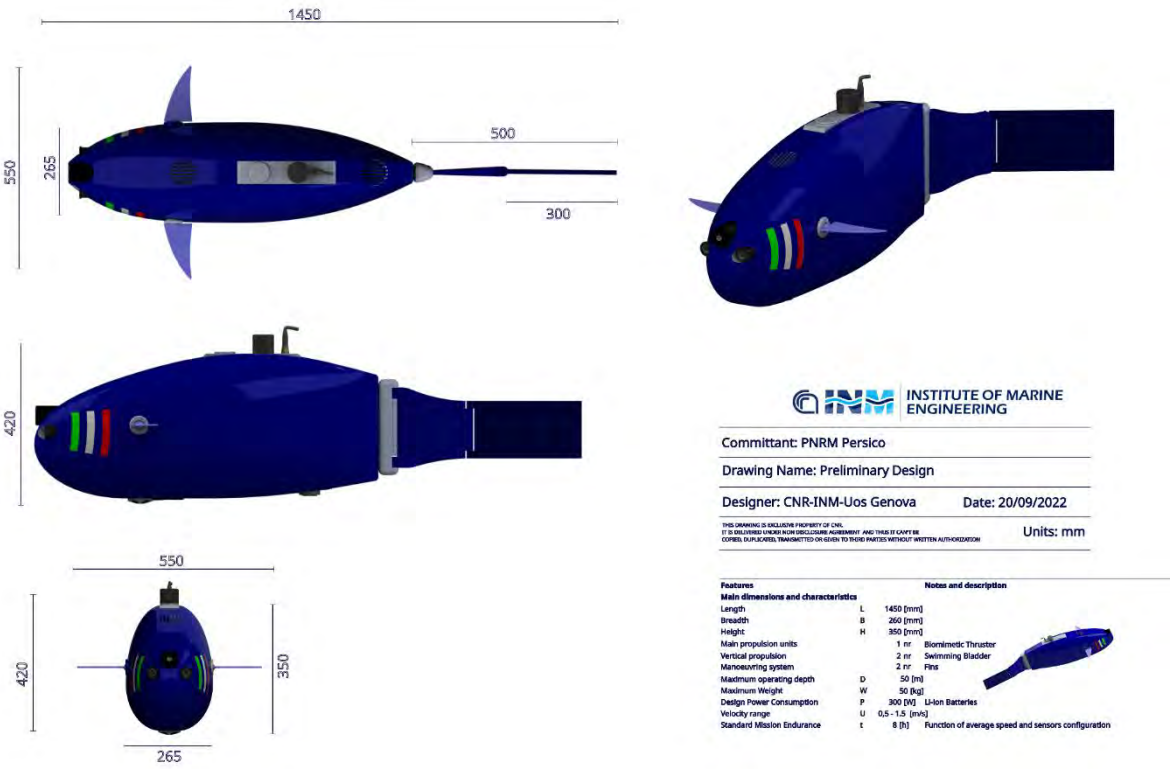


Fig.7: Final design of the prototype

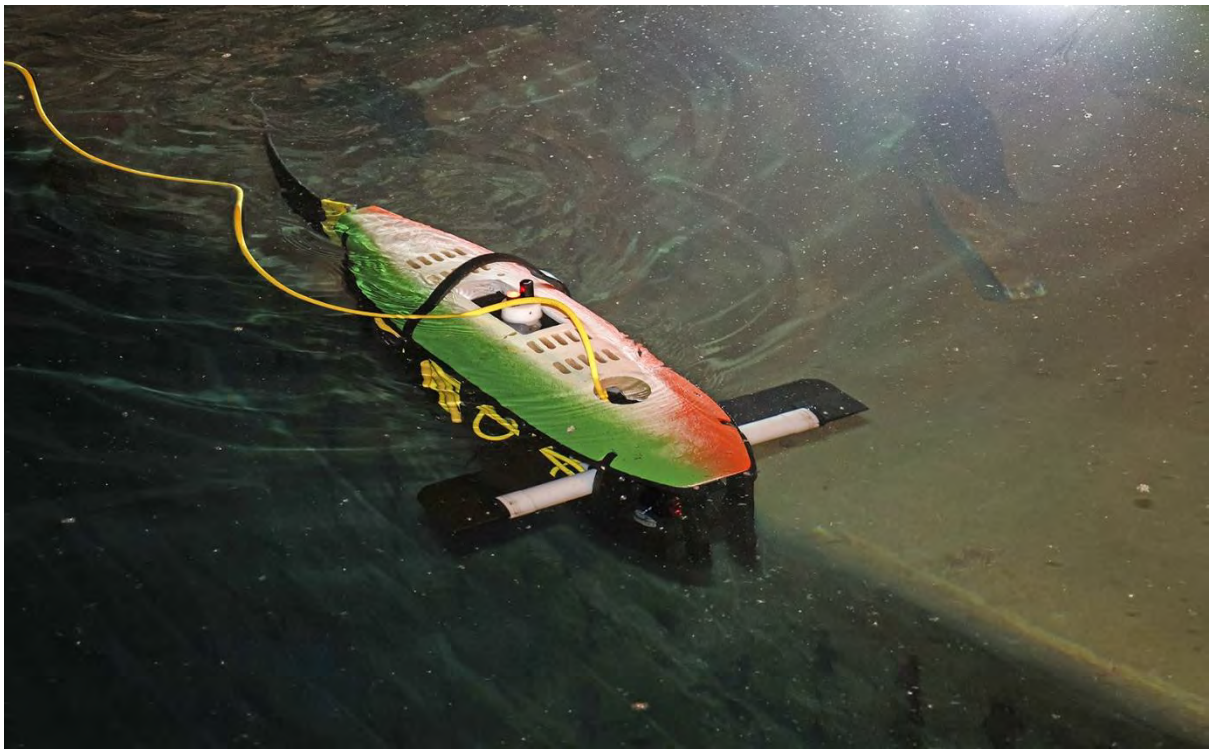


Fig.8: The robotic fish during trials

6. Capabilities

The moment of truth always comes when the prototype is deployed in water for the first time. An extended phase for initial functional test and identification of the basic motion characteristics is mandatory to provide the essential information on top of which is then possible to develop all the

advanced autonomous features that the platform will exploit in full operational scenarios. Four main motion characteristics have been studied:

1. the propulsion generation, provided by the oscillatory motion of the main fin;
2. the change of the horizontal motion direction, obtained by biasing the oscillation of the main fin;
3. the heave motion controlled by the bladders' volume change;
4. the vertical motion pattern provided by the lateral fin configuration.

6.1. Propulsion generation

The generation of the propulsion along the surge axis of the platform is provided by commanding a suitable oscillatory motion to the main fin (note: at the present time, the main fin is composed by a single moving joint, a secondary joint is scheduled to be mounted and tested in the next advancement of the project). The oscillatory motion of the fin is provided commanding the actuation motor with a signal, characterized by the two parameters, namely the amplitude and the frequency of the oscillation. In particular, the signal is a piecewise step function that acts as a reference rotation rate for the actuation motor: the motor is set to rotate at a constant reference speed until on end of the fin motion range is reached, then the value is instantly inverted so that the fin will move all the other way around until the other motion limit is reached. The velocity value (ω) is computed so that the fin motion is a quasi-sinusoidal wave with the desired amplitude (A) and frequency (f) as depicted in Fig.9.

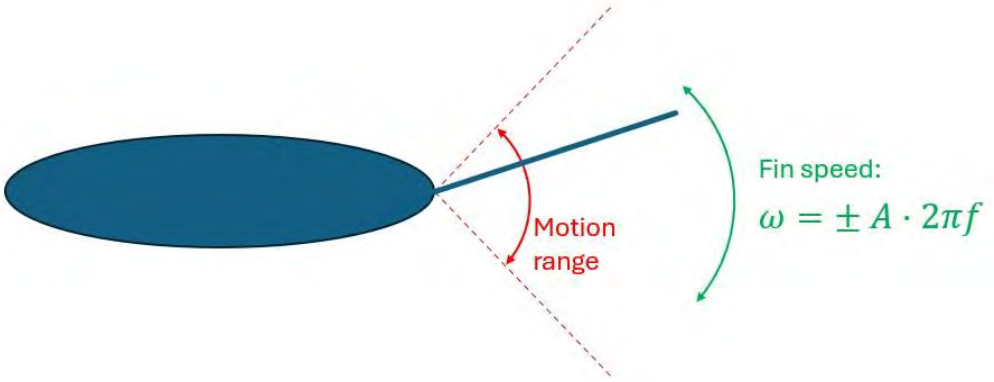


Fig.9: Generation of the rotation rate signal of the fin

The test and identification phase of the surge motion generation (in particular the measurement of the surge speed reached under the propulsion force provided by the fin) is obtained by means of commanding the fin actuation with a scheduled sequence of setting for both the amplitude (in the range: 5 – 35°) and frequency (in the range: 0.5 – 2.5 Hz).

As reported in Fig.8, the maximum speed reached is of about 0.25 m/s, definitely not a high speed, but reasonable given the (still) semi-open from (the final version of the prototype is intended to have a protective soft "skin", furtherly reducing the drag coefficient), and reduced aspect ratio of the fin (actually a rectangle of dimensions: 0.51 m x 0.15 m).

On the basis of such "fin-to-speed" measures, it is possible to design a mathematical model for the propulsion estimation, which will also ease the implementation of proper surge speed regulators for autonomous operations. The model, obtained via data-based least square method, is the following:

$$\dot{u} = k_{u|u|}\, u|u| + k_{|r|}\, |r| + b_u\, A \cdot f$$

$$k_{u|u|} = -1.0785 \quad , \quad k_{|r|} = -0.0107 \quad , \quad b_u = 0.0018$$

where u is the surge speed, r is the yaw-rate, A and f are the fin motion amplitude and frequency

respectively, each term is multiplied by the corresponding identified parameter. With respect to Fig.10, it can be noticed the good super-imposition of the estimated surge speed (green line) with the actual measured value (blue line).

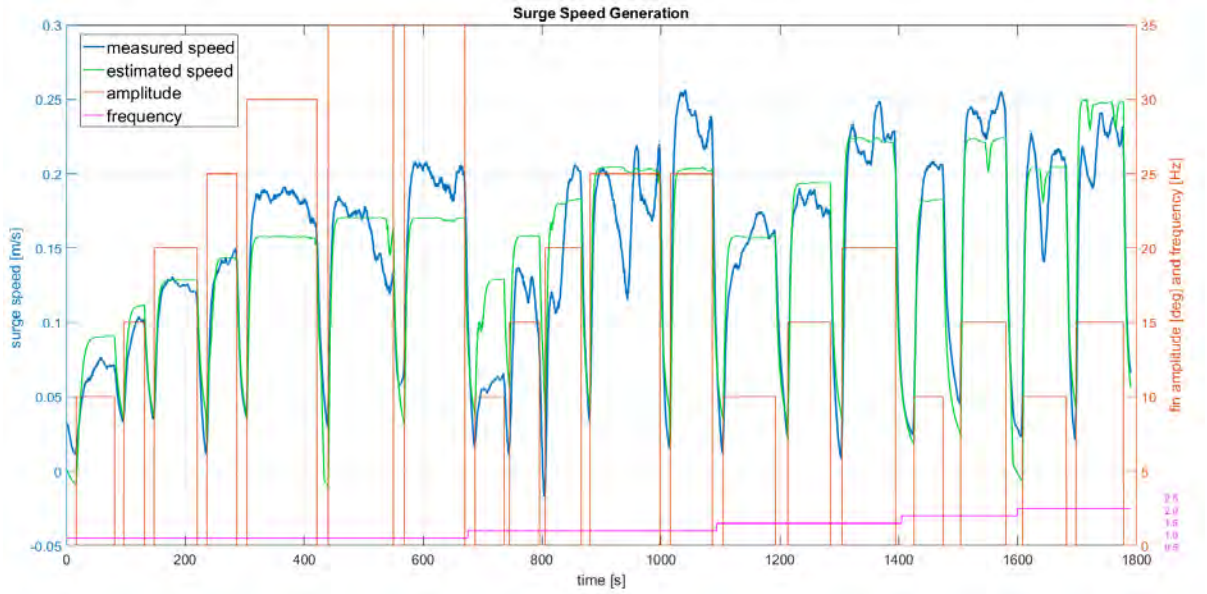


Fig.10: Results of the surge motion measurement under fin-provided propulsion

6.2. Horizontal motion direction

The horizontal direction of motion is obtained by offsetting the oscillatory motion of the fin from the platform longitudinal centerline, thus creating a lateral force component that reflects into a rotational torque that steers the platform on the horizontal plane.

Due to the respectively low weight of the prototype itself with respect to the actual force generated by the fin, as result of the reacting moment, the head of the fish produces a slight oscillatory motion that is captured by inertial sensor, mounted onboard, used to measure the direction of the platform. For such a reason, a pre-filtering of such a heading measure has to be processed in order to average the measurement producing a properly stable value of the actual direction of motion. The comparison between raw vs. filtered heading signal is reported in Fig.11, where the oscillation of about 5 – 6° (peak-to-peak) is noticeable.

With the same procedure employed for the surge model identification, the yaw motion characteristics is identified, producing the following mathematical model:

$$\dot{r} = k_r\, r + b_r\, u \cdot \delta$$

$$k_r = -0.0992 \quad , \quad b_r = 0.0039$$

where r is the yaw-rate, u the surge speed, δ the input direction of steering, and the respective drag and input coefficients.

Fig.12 reports (in the top plot) the yaw-rate raw signal (blue line), with the super-imposition of the estimated measure (green line), in function of the command steering direction. The actual surge speed is also reported in the bottom plot of Fig.12.

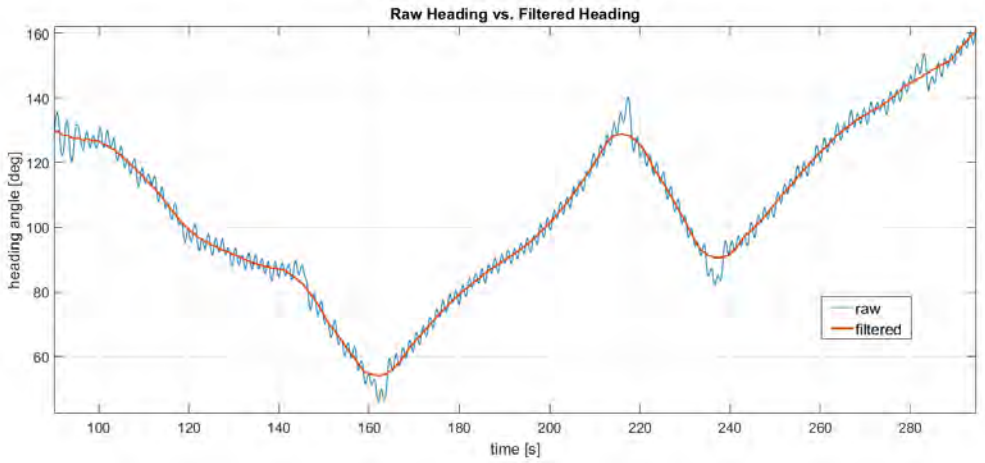


Fig.11 – The heading oscillation produced by the fin motion

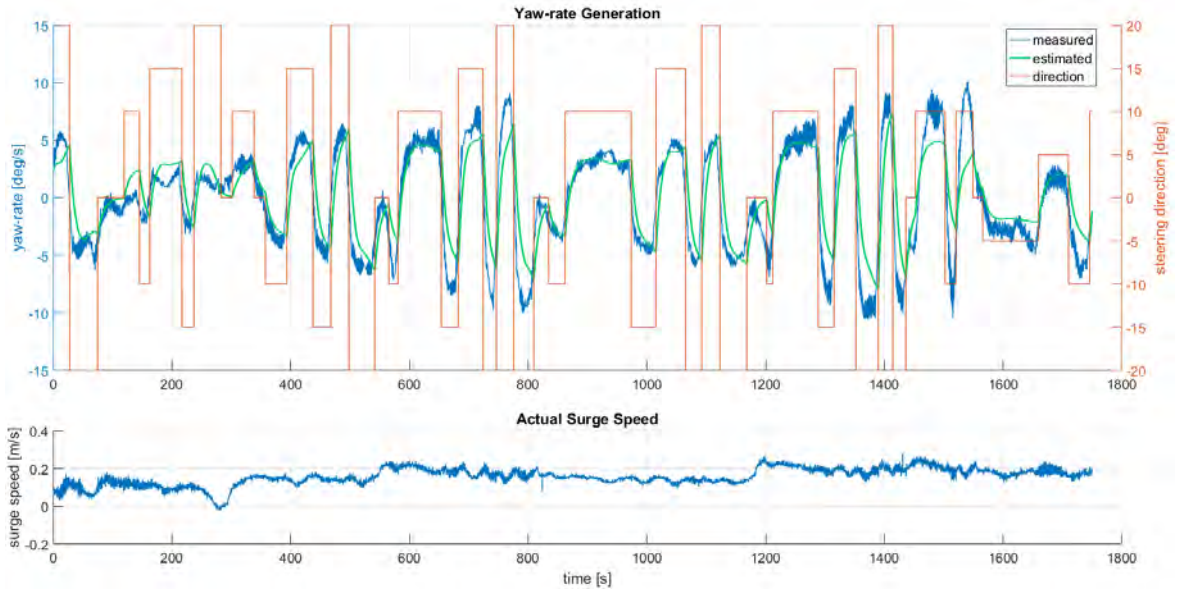


Fig.12: Results of the yaw motion measurement

6.3. Heave motion with bladders

The vertical or heave motion can be produced in any condition of the vehicle (moving or resting) through the proper actuation of the two-bladder system installed in the inner part of the platform. By changing the internal volume of the bladders, the buoyancy force is varied thus producing a force that moves the platform along the vertical direction. The variation of the vertical position can be measured by means of the pressure sensor, which directly provides the depth reading, and by the altitude from the sea-bottom measured by the DVL sensor.

Such measurements allow to identify the heave characteristics, obtaining the following model:

$$\dot{w} = k_{w|w|}\, w|w| + B_o + b_b\, f_b$$

$$k_{w|w|} = -3.0707 \quad , \quad B_o = -0.0003 \quad , \quad b_b = -0.0002$$

with w the heave speed, B_o the buoyancy offset (the platform is configured to be neutral, but naturally a small weight bias will be always present; for instance, the presence of bubbles can alter the buoyancy

equilibrium), f_b the bladder input (intended as the percentage of air/water ratio in the bladders volume: 100% full air, -100% full water), and the respective parameters.

The captured heave speed measurements are compared with the estimated values in Fig.13 (top plot), in response to the bladder commands; the bottom plot of Fig.13 reports the depth profile assumed during the trial. It has also to be noted that, since the buoyancy offset can vary from one operation to the other (because of environmental condition, slight payload modifications, or other unpredictable factors), an augmented estimation scheme can be implemented in order to on-line adapt the value, also to keep that into account for automatic depth/altitude regulation.

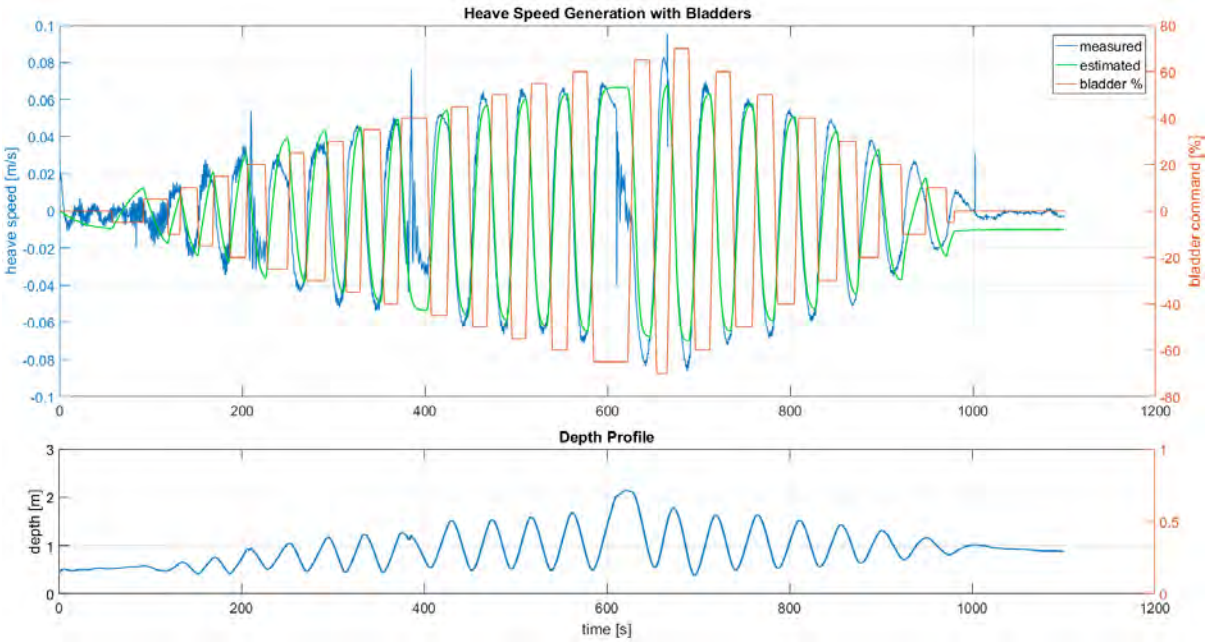


Fig.13: Results of the heave motion measurement with bladder actuation

6.4. Heave motion with lateral fins

When the vehicle is subject to the longitudinal propulsion (i.e. the surge speed is $u > 0$), the heave motion can be also controlled by means of the lateral fins' actuation. Such fins act as the canard wings of airplanes and the proper regulation of their angle of attack, with respect to the surge flow, has the effect of producing a pitching torque which reflects on an attitude change, in turn leading to a depth change.

Thus, the heave dynamic model can be enhanced by adding the term related to the lateral fins' force component, with f_f the angle of attack of the fins, u the surge speed and b_f the input coefficient, all the other terms are as mentioned in the previous section, obtaining the following form:

$$\dot{w} = k_{w|w|}\, w|w| + B_o + b_b\, f_b + b_f\, u \cdot f_f$$

$$k_{w|w|} = -3.0707 \quad , \quad B_o = -0.0003 \quad , \quad b_w = -0.0002 \quad , \quad b_f = -0.0252$$

An exemplificative behavior test of the fin-based heave motion is shown in Fig.14, where the depth change is measured in function of the fin position variation (top plot); the estimated value of the heave speed is computed by means of the mathematical model. The bottom plot of Fig.14 reports the depth profile during the test.

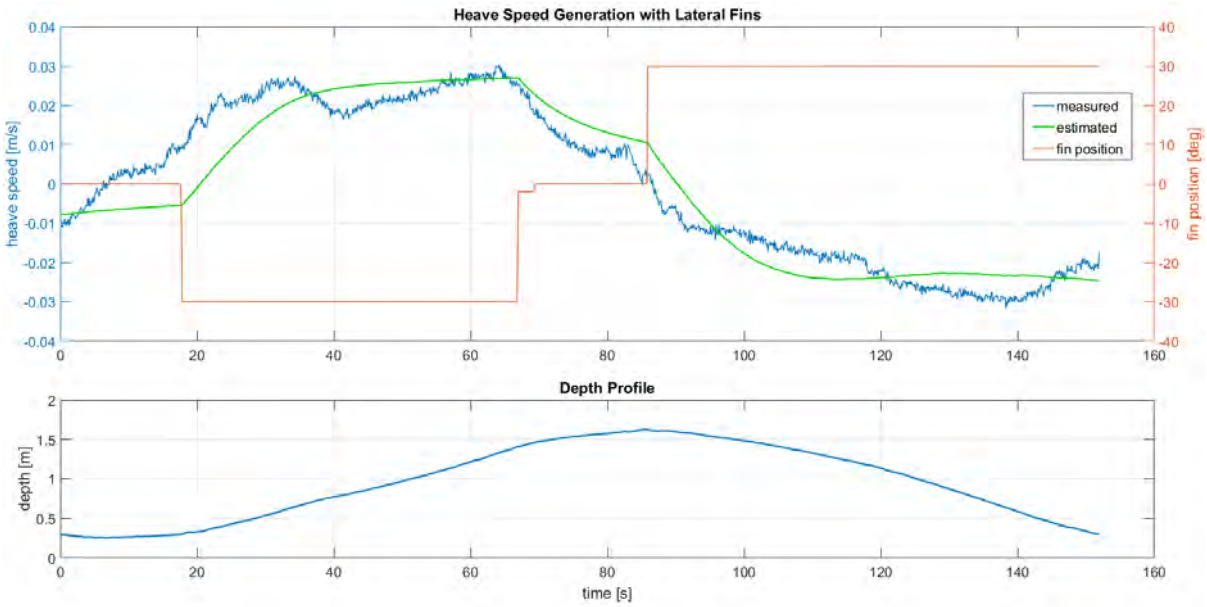


Fig.14: Results of the heave motion measurement with lateral fin actuation

6.5. The robotic fish in action

This section simply shows, through a couple of image sequences, the robotic fish in action during motion tests in a towing tank facility, Fig.15, and open waters, Fig.16.

Fig.15: Image sequence from below of the robotic fish in motion

Currently, the prototype is operated through manual command, sending proper control signal by means of the umbilical cable that allows a continuous control and monitoring of the system (and, of course, a physical mean to limit the motion range and retrieve the platform in case of malfunction). Although the slow dynamics exploited by the platform, in terms of maximum motion velocities, this characteristic does not prevent the prototype to be effective and reliable from the exploration and data gathering standpoints. Moreover, one of the key features of the project strongly relies on the development of a silent platform with respect to classical underwater vehicles. Such objective has certainly been reached, guaranteeing in parallel satisfactory operational capabilities.

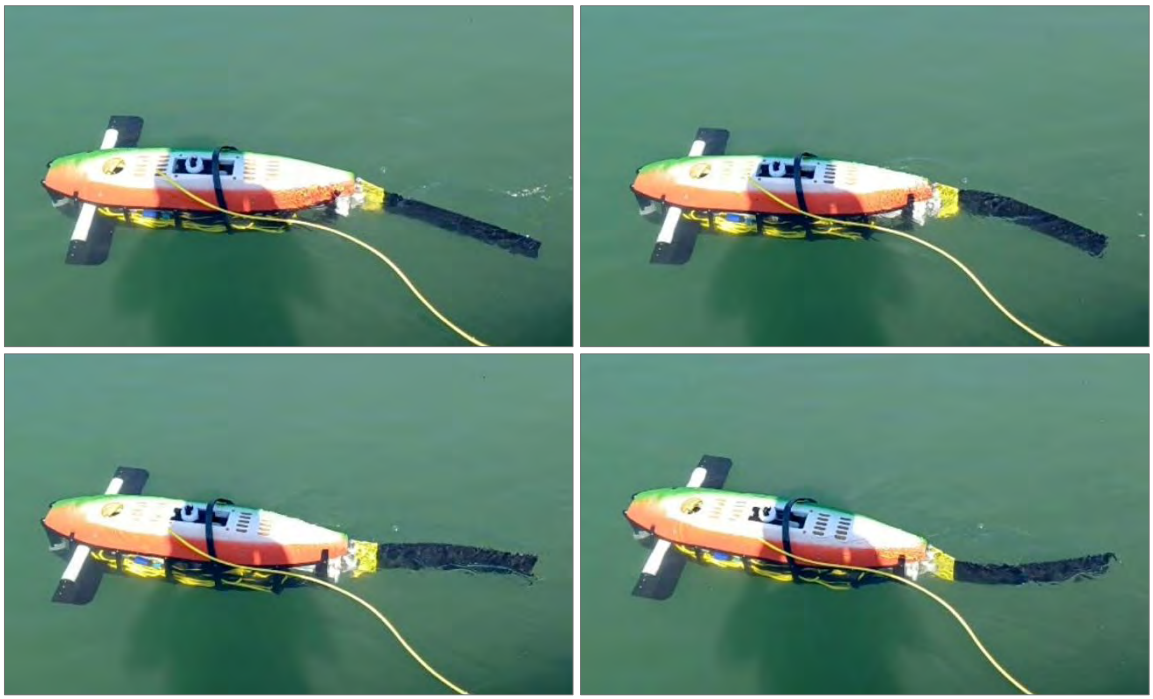


Fig.16: Image sequence from above of the robotic fish in motion

7. Future steps

The PERSICO project has reached and proven the objective of developing an operational prototype of a robotic fish, able to exploit exploration and data gathering capabilities, matching the challenging and innovative objective of providing a silent platform for underwater operations.

The project is currently active, with one final year of development, improvements and fine-tuning of the system, in order to enhance the capability of the platform.

One key objective is the design and implementation of suitable automatic controls and advanced guidance modules, which are fundamental to provide the autonomous capabilities to operate in operational scenarios. This phase is currently in progress; given the successful identification of the dynamic models (described in the previous sections), it is possible to design effective regulation scheme to provide controls for: *i*) the surge speed (i.e. propulsion regulation); *ii*) the heading (i.e. guidance in the operational scenario); *iii*) the depth/altitude, allowing the transition along the vertical underwater domain.

The removal of the umbilical cable is a necessary step in order to multiply the space coverage during real operations. The power supply is not an issue, given the presence of the battery onboard the vehicle. The cable is only used for fast data transfer and the idea is to substitute it by installing an acoustic communication system (composed by a beacon mounted onboard the platform and an acoustic head positioned at the operator station). The acoustic communication system allows to send commands to the platform and receive basic information set; because of the acoustic physics, the communication bandwidth is drastically reduced (with respect of usual cabled or in-air-wifi connections) and so the amount of transferred data has to be carefully defined. Lastly, the acoustic communication system allows to measure the relative position of the platform, in the underwater domain, providing a more reliable location sensing (to be used by the guidance system) with respect to dead-reckoning navigation approaches (i.e. estimating the position by calculating movement by means of velocity and attitude integration) that tend to numerically diverge in short time.

The development of the above-mentioned objectives has the aim of drastically improving the autonomy of the robotic fish in terms of extended area coverage, autonomous decision-making during operations,

optimization of energy usage, low acoustic presence.

A particular attention is posed on this latter aspect and, for this reason, a thorough study and measurement phase of the acoustic footprint has been scheduled to characterize the irradiated noise in the environment and the possibility of acoustic-based detection and tracking.

Acknowledgements

PERSICO (PEsce Robotico SIlenzioso per Campionamento e Osservazione - Silent Robotic Fish for Observation and Sampling) is a project co-funded by the Italian National Program of Military Research (PNRM).

References

BIBULI, M.; ODETTI, A.; CACCIA, M.; PRESICCI, C.; FERRETTI, R.; ARACRI, S. (2022), *Networked Robots and IoT: Real Time Data acquisition at its finest*, COMPIT Conf., Pontignano

JUNZHI, Y.; TAN, M.H.; WANG, S.; CHEN, E. (2004), *Development of a biomimetic robotic fish and its control algorithm*, IEEE Trans. Systems, Man, and Cybernetics, Part B (Cybernetics), 34/4, pp.1798-1810

LI, Y.; XU, Y.; WU, Z.; MA, L.; GUO, M.; LI, Z.; LI, Y. (2022), *A comprehensive review on fish-inspired robots,* Int. J. Advanced Robotic Systems 19(3)

ODETTI, A.; BRUZZONE, G.; ALTOSOLE, M.; VIVIANI, M.; CACCIA, M. (2020), *SWAMP, an Autonomous Surface Vehicle expressly designed for extremely shallow waters*, Ocean Eng. 216

ODETTI, A.; BIBULI, M.; BRUZZONE, G.; CACCIA, M.; SPIRANDELLI, E.; BRUZZONE, G. (2017), *e-URoPe: a reconfigurable AUV/ROV for man-robot underwater cooperation*, 20th IFAC World Congress, Toulouse

RUS, D.; TOLLEY, M.T. (2015), *Design, fabrication and control of soft robots*, Nature 521, pp.467-475

SFAKIOTAKIS, M.; LANE, D.M.; DAVIES, J.B.C. (1999), *Review of Fish Swimming Modes for Aquatic Locomotion*, IEEE J. Oceanic Eng. 24(2), pp.237-252

ZHANG, Z.; WANG, Q.; ZHANG, S. (2024), *Applications of biomimetic propulsion for underwater vehicles*, Biomimetics

Navigating AI in Naval Architecture: A Comparative Effectiveness Study of Machine Learning Models for Ship Stability

Karolina Bierkowska, NTNU, Ålesund/Norway, karolina.bierkowska@ntnu.no
Henrique Murilo Gaspar, NTNU, Ålesund/Norway, henrique.gaspar@ntnu.no
Tomasz Hinz, Gdańsk Tech/Deltamarin, Gdańsk/Poland, tomhinz@pg.edu.pl

Abstract

This study comparatively analyses diverse AI/ML models on an established dataset of hull variants and Second-Generation Intact Stability Criteria metrics, a time-consuming task in the early stages of ship design. This selection encompasses diverse AI techniques, each recognised for its unique strengths, featuring Artificial Neural Networks, Decision Trees, Probabilistic Models, and Large Language Models. In this article, we focus primarily on one of the failure modes: excessive acceleration. This study benchmarks the models against each other in terms of predictive accuracy, computational efficiency, robustness. The objective is to evaluate the analyses and identify a "superior" AI/ML strategies for faster, more reliable early-stage design stability assessment. This research guides naval architects in selecting suitable emergent AI tools to enhance design space exploration, ultimately contributing to filter the current AI "hype" into useful NA practices in the industry.

1. Introduction

1.1. Motivation for Using SGISC in This Study

Ship stability has a significant impact on operations, including safety, efficiency, and regulatory compliance. Between 1998 and 2014, nine documented accidents occurred due to failures in dynamic stability systems.

In October 1998, the APL China lost approximately 400 containers overboard and damaged another 400 due to parametric rolling, *France et al. (2003)*. In September 2008, aboard the Chicago Express, one person died and several were injured when the ship rolled to an angle of 45 degrees due to excessive acceleration, *Kaufmann (2009)*. In January 2012, the Rabaul Queen sank, reportedly due to broaching-to, *Australia and Mininga (n.d.)*. These incidents represent a subset of documented cases. Of the nine accidents, eight were compliant with the 2008 IS Code, *IMO (2009)*, the current intact stability criteria—classified as safe in terms of stability. The underlying cause of each accident was related to stability, highlighting that the current criteria do not sufficiently address all relevant failure modes. Based on these findings, the development of new stability criteria was initiated.

In 2020, the IMO's Maritime Safety Committee approved the interim guidelines for the new stability criteria, with the first explanatory notes published in April 2023, *IMO (2020,2023)*. A first revision of the document was published in 2024, *IMO (2024)*. Although the criteria are not yet mandatory, designers are encouraged to consider them when designing new ships, *Begovic et al. (2023)*.

These new criteria are known as the Second Generation Intact Stability Criteria (SGISC). They are not intended to replace the 2008 IS Code, but rather to complement it. SGISC addresses five stability failure modes: parametric rolling, pure loss of stability, broaching-to, dead ship condition, and excessive acceleration. SGISC is based on a multi-level framework, Fig.1, where Level 1 is the most conservative but also the easiest to calculate, *IMO (2024)*. The second level is based on solving the differential equation of a simplified model of ship motion.

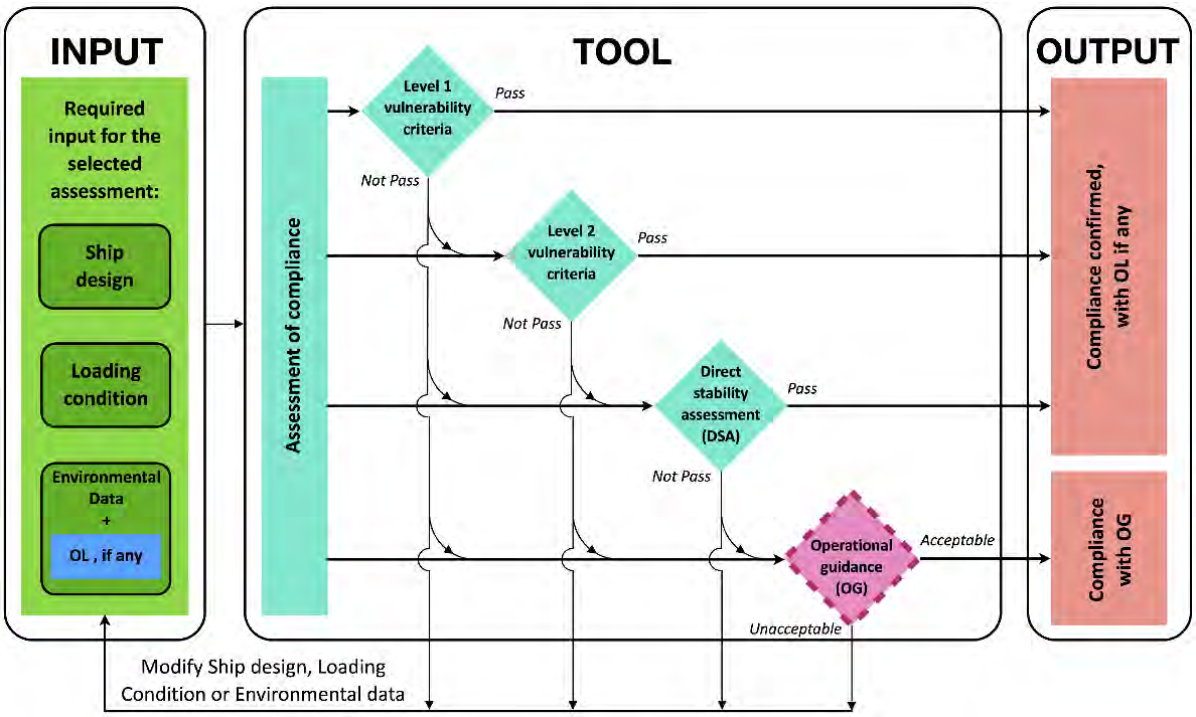


Fig.1: Scheme of SGISC framework, *Petacco and Gualeni (2020)*

1.2. Motivation for Using AI

To calculate SGISC excessive acceleration, you need information such as main dimensions and hull shape, bilge keel geometry, locations of passengers and crew, and loading conditions. These design parameters are often unavailable during the early stages of ship design, while at later stages, main dimensions and other characteristics are difficult or impossible to modify. Therefore, it is crucial to evaluate this criterion before deciding on the main dimensions, as they significantly affect the outcome, *Sayli et al. (2007).* Setting up the required parameters for this criterion takes approximately 15 minutes. Even when all required dimensions are available, evaluating multiple design options using AI is significantly faster, often requiring only seconds.

AI has already demonstrated its applicability in ship design. For example, it can be used to predict GM and KG at the preliminary stage, *Alkan et al. (2004)*, to optimize trim, *Vasilev et al. (2024)*, or to develop hull form, *Bagazinski and Ahmed (2023), Khan et al. (2023), Ichinose and Gaspar (2023).*

2. Data Preparation for AI Models

This study focuses on the failure mode of excessive acceleration, evaluated at Level 2 of the SGISC framework. The criterion was calculated using the NAPA software, and the hull geometry was generated through a parametric model in CAESES, *Harries et al. (2019)*. Further details on the parametric model preparation and calculation process can be found in previous work, *Bierkowska et al. (2025).*

The models were prepared according to the input configurations listed in Table I. The data were grouped based on dimensional parameters, and for each input set, the worst-case outcome was selected. Specifically, if any combination within a set (with identical dimensional values) failed the criterion (A01 = 0), the entire set was marked as failing. This data preparation method ensures that the models are trained using the most safety-critical cases, making their predictions more conservative. The dataset was then split into a training set (90% of the data) and a test set (10%).

Table I: Inputs

Model name	input
Model.1	LOA, Beam, Height, CB
Model.2	LOA, Beam, Height, CB, Length Keel, Breadth Keel, GM
Model.3	LOA, Beam, Height, CB, Length Keel, Breadth Keel, GM, X Point, Z Point
Model.4	LOA, Beam, Height, CB, LCB
Model.5	LOA, Beam, Height, CB, Length Keel, Breadth Keel, GM, LCB
Model.6	LOA, Beam, Height, CB, Length Keel, Breadth Keel, GM, X Point, Z Point, LCB
Model.7	LOA, Beam, Height, CB, LCB, T
Model.8	LOA, Beam, Height, CB, Length Keel, Breadth Keel, GM, LCB, T
Model.9	LOA, Beam, Height, CB, Length Keel, Breadth Keel, GM, X Point, Z Point, LCB, T
Model.10	LOA, Beam, CB, Length Keel, Breadth Keel, GM
Model.11	LOA, Beam, CB, Length Keel, Breadth Keel, GM, X Point, Z Point
Model.12	LOA, Beam, CB, LCB
Model.13	LOA, Beam, CB, Length Keel, Breadth Keel, GM, LCB
Model.14	LOA, Beam, CB, Length Keel, Breadth Keel, GM, X Point, Z Point, LCB
Model.15	LOA, Beam, CB, LCB, T
Model.16	LOA, Beam, CB, Length Keel, Breadth Keel, GM, LCB, T
Model.17	LOA, Beam, CB, Length Keel, Breadth Keel, GM, X Point, Z Point, LCB, T

3. Using AI to Evaluate SGISC Stability

3.1. Deep Neural Network

TensorFlow, *Shanmugamani (2018)*, was used to train the Deep Neural Network (DNN) models. All models were trained using the same architecture. Table II presents the number of neurons in each layer for different iterations. A normalization layer was placed after the input layer. After each hidden layer, a dropout layer, *Salehin and Kang (2023)*, with a rate of 0.2 was added.

Table II: Number of neurons on layers for different iteration

Iteration number	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5	Layer 6	Layer 7	Layer 8
DNN-0	32	16	8					
DNN-1	64	32	16	8				
DNN-2	16	8						
DNN-3	128	64	32	16	8			
DNN-4	256	128	64	32	16	8		
DNN-5	512	256	128	64	32	16	8	
DNN-6	1024	512	256	128	64	32	16	8

The Rectified Linear Unit (ReLU), *Panda (2023)*, was used as the activation function. The output layer uses a sigmoid activation function, which produces values between 0 and 1. Predictions greater than 0.5 were classified as 1 (pass), while those less than or equal to 0.5 were classified as 0 (fail). Adaptive Moment Estimation (Adam), *Kingma and Ba (2014)*, was used as the optimizer. The binary Cross-Entropy function was employed as the loss function. Model performance was evaluated using the binary accuracy metric. 25% of the training data was used for validation. The model exhibiting the lowest validation loss was selected for evaluation.

3.2. Large Language Models

For the Large Language Models (LLM) experiments, Ollama, https://github.com/ollama/ollama, was used, specifically the phi3:instruct model, https://ollama.com/library/phi3:instruct. The phi3:instruct was not trained; instead, it was used through prompt-based queries. In each prompt, n examples from the training set that were similar to the test case were included. These examples were selected based

on the smallest Euclidean distance to the test vector, after standardizing the features. Phi3:instruct was asked to predict whether the test case met the stability criterion. To determine the optimal configuration, the models were tested with *n* values of 2, 10, and 20. For each model, 100 test examples were randomly selected to determine which n value produced the best results. The full test set was then evaluated using that optimal n. The value of n used for each model is shown in Table III.

Table III: Number of examples in prompt

model	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
n	10	2	10	2	20	2	20	2	2	10	20	10	20	10	10	20	2

3.3. Decision Tree and Gradient Boosting

Another machine learning model used in these studies is Decision Trees. Decision Trees, *Watt et al. (2020)*, are among the most commonly used models in supervised learning, offering an intuitive and interpretable framework for classification tasks. Decision Tree induction is typically performed through recursive partitioning of the input space, guided by criteria such as information gain, gain ratio, or the Gini index, *Yale et al. (2017)*, which determine the attribute that best separates the data at each step. In this part of the analysis, the library used to create the Decision Tree was the Scikit-learn library, *Pedregosa et al. (2011)*. An example of a Decision Tree is shown in Fig.2.

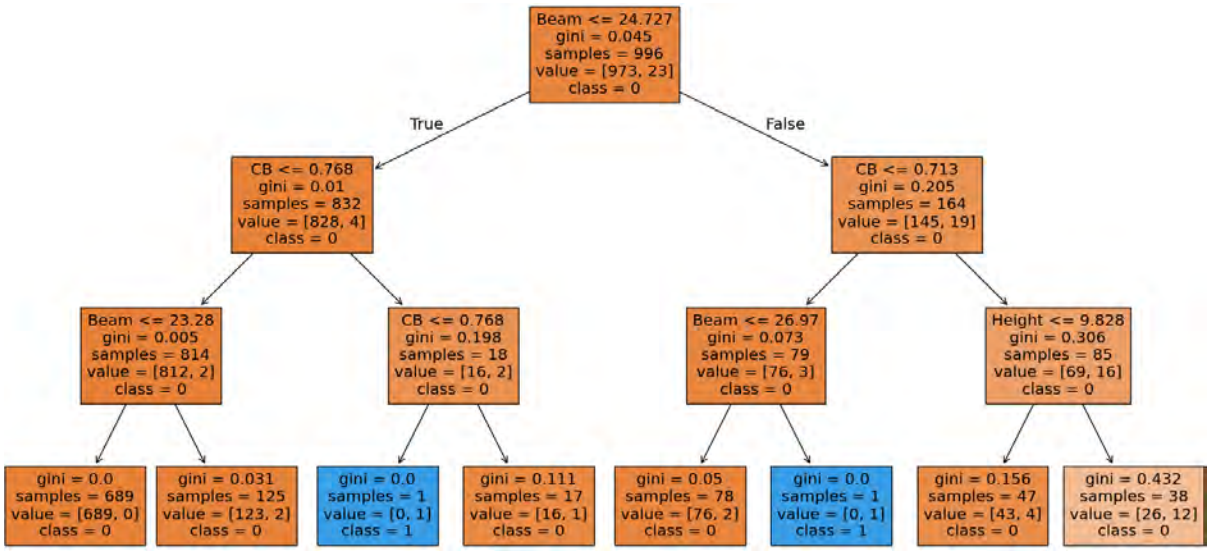


Fig.2: Example of Decision Tree (Model 1, max depth 3)

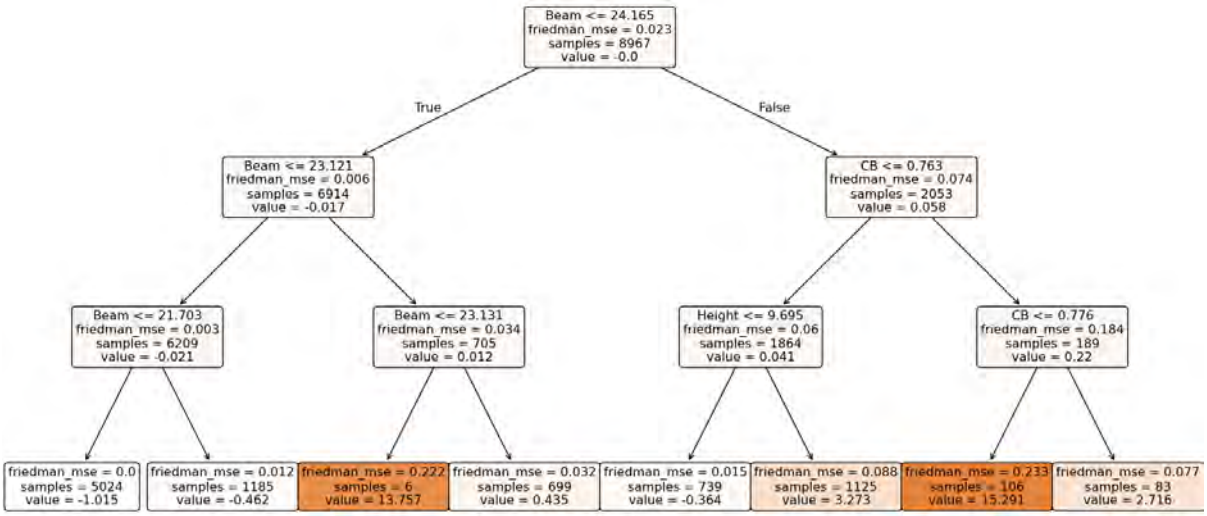


Fig.3: Example of one tree from Gradient Boosting (Model 1, max depth 3)

While a single Decision Tree offers interpretability and simplicity, its instability and tendency to overfit have motivated the development of ensemble approaches Gradient Boosting is an ensemble learning technique that builds predictive models in a stage-wise, sequential manner, where each new decision tree is trained to reduce the errors (residuals) of the previous ensemble. Gradient Boosting, *Friedman (2001)*, optimizes a chosen loss function using gradient descent, allowing it to achieve high predictive accuracy—though at the cost of greater risk of overfitting and higher computational demand. The scikit-learn library was used again. Example of Gradient Boosting is shown in Fig.3.

3.4 Probabilistic programming

Probabilistic programming provides a framework for specifying complex statistical models in a high-level, declarative manner, enabling automated inference over probabilistic models, *Blei et al. (2017)*. Unlike traditional machine learning approaches, which often yield point estimates, probabilistic programming embraces uncertainty by treating model parameters as random variables with associated probability distributions, *Ness (2025)*. Within this paradigm, Bayesian Neural Networks (BNNs) Pinheiro Cinelli et al. (2021) extend conventional neural networks by placing prior distributions over their weights and biases, resulting in posterior distributions. One of the most widely used libraries in probabilistic programming is PyMC, *Abril-Pla et al. (2023)*. Example of probabilistic programming model is shown in Fig.4.

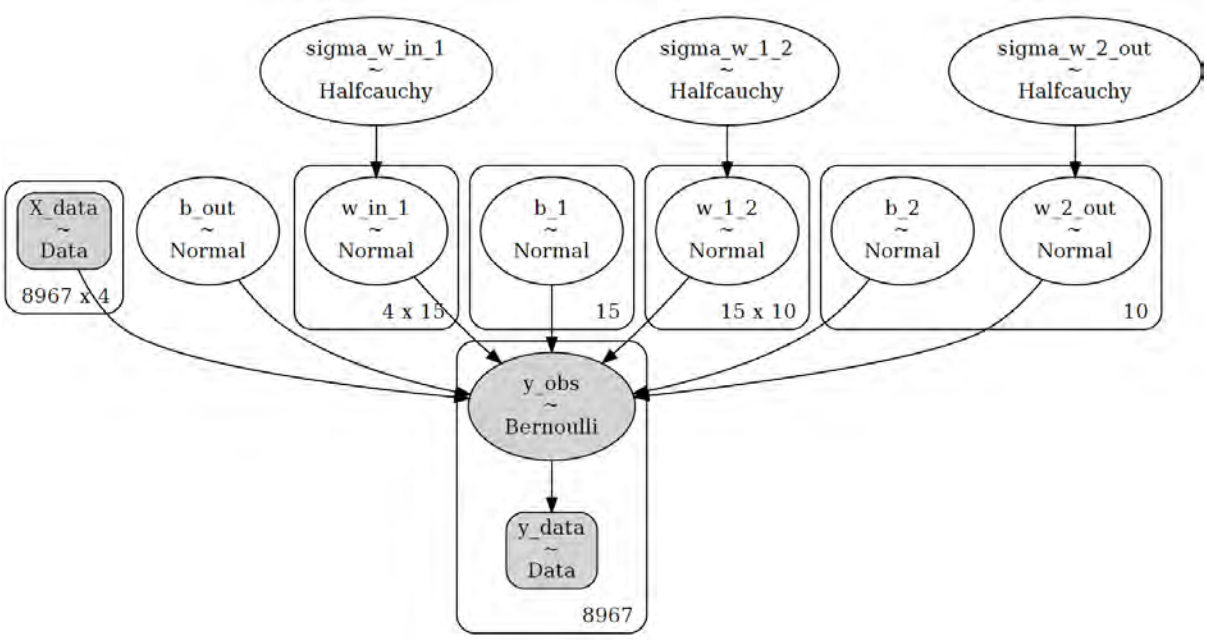


Fig.4: Example of probabilistic programming

4. Matthews Correlation Coefficient metrics

Some models are imbalanced—for example, some have only 2% of cases where the criterion is passed—accuracy is not a reliable metric. Therefore, the Matthews Correlation Coefficient was used instead.

The Matthews Correlation Coefficient (MCC), *Chicco and Jurman (2020)*, is a measure of how well a binary classification model performs. An MCC of +1 indicates perfect prediction, 0 indicates random guessing, and −1 indicates complete misclassification. It is particularly useful for imbalanced datasets. MCC is defined as:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Where:
- True Positive (TP) – both actual and predicted values are 1
- True Negative (TN) – both actual and predicted values are 0
- False Positive (FP) – actual value is 0, but predicted as 1
- False Negative (FN) – actual value is 1, but predicted as 0

In cases where the phi3:instruct failed to provide a prediction—likely due to hallucination—these were treated as False Positives if the actual value was 0, and as False Negatives if the actual value was 1.

5. Results

MCC scores for phi3:instruct, Probabilistic programming, Decision Tree and Gradient Boosting are presented in Table IV. MCC scores for the DNNs are presented in Table V.

Table IV: MCC score for phi3:instruct, Probabilistic programming, Decision Tree and Gradient Boosting

model	phi3:instruct	Probabilistic programming	Decision Tree	Gradient Boosting
Model.1	0.211	0	0.146	0.061
Model.2	0.485	0	0.815	0.742
Model.3	-0.989	0	0.872	0.774
Model.4	0.232	0	0.489	0.202
Model.5	-0.785	0	0.899	0.778
Model.6	-0.983	0	0.837	0.796
Model.7	0.002	0	0.733	0.321
Model.8	0.544	0	0.907	0.786
Model.9	-0.811	0	0.902	0.805
Model.10	0.503	-0.007	0.765	0.731
Model.11	-0.992	0	0.864	0.769
Model.12	0.264	0	0.582	0.376
Model.13	-0.917	0	0.872	0.767
Model.14	-0.947	0	0.895	0.794
Model.15	0.362	0	0.693	0.321
Model.16	-0.918	0.004	0.896	0.782
Model.17	-0.926	0	0.898	0.802

Table V: MCC score for Deep Neural Networks

model	DNN-0	DNN-1	DNN-2	DNN-3	DNN-4	DNN-5	DNN-6
Model.1	0	0	0	0	0.206	0	0
Model.2	0.64	0.667	0.63	0.698	0.723	0.782	0.831
Model.3	0.643	0.668	0.619	0.675	0.735	0.822	0.849
Model.4	0.206	0.367	0	0	0	0	0.206
Model.5	0.693	0.744	0.645	0.808	0.893	0.926	0.945
Model.6	0.658	0.714	0.635	0.801	0.781	0.921	0.944
Model.7	0.51	0.559	0.206	0.585	0.596	0.663	0.581
Model.8	0.722	0.82	0.662	0.897	0.943	0.962	0.971
Model.9	0.708	0.772	0.654	0.871	0.918	0.953	0.956
Model.10	0.637	0.653	0.626	0.67	0.67	0.692	0.709
Model.11	0.646	0.653	0.621	0.665	0.682	0.704	0.688
Model.12	0.358	0.367	0.292	0.361	0	0	0.161
Model.13	0.668	0.725	0.646	0.794	0.839	0.882	0.892
Model.14	0.659	0.691	0.618	0.785	0.853	0.897	0.914
Model.15	0.409	0.495	0.142	0.596	0.585	0	0.495
Model.16	0.699	0.75	0.663	0.846	0.901	0.938	0.946
Model.17	0.695	0.742	0.626	0.825	0.883	0.939	0.94

The plot of MCC scores for all models is presented in Fig.5. For clarity, only the DNN results corresponding to the neuron configuration that produced the best performance are presented.

Fig.5: The plot of MCC score for all models

6. Discussions

This study explores the potential of using various AI techniques in early-stage ship design to accelerate the calculation process. This approach is particularly useful when not all required dimensions are available, or when designers wish to test multiple sets of dimensional parameters.

For example, predictions using phi3:instruct take approximately 5 seconds, while DNN models require only a few milliseconds. In contrast, calculations performed in NAPA software take around 15 minutes.

The results demonstrate promising performance, particularly for DNNs and Decision Trees with MCC values exceeding 0.9. The best-performing DNN model achieved an MCC of 0.971.

In contrast, Probabilistic programming assigned the same class to all inputs, demonstrating it ineffective for this application.

Phi3:instruct, on the other hand, exhibited poor scores (close to -1) likely caused by hallucinations. Using alternative models from Ollama may help decrease the frequency of these hallucinations.

If its performance improves, phi3:instruct could serve as a viable alternative for users without AI expertise—which was the primary motivation for including it in this study. It does not require a large dataset, is open-source, and operates offline, ensuring that no data is transmitted to external servers.

Future work could investigate whether using examples with varying combinations of dimensional parameters in the phi3:instruct prompt affects prediction accuracy. For instance, one prompt might include examples with bilge keel dimensions, while another uses position-related parameters such as X Point and Z Point. This is a realistic approach, as in practice we may not always have access to a complete set of data from previous projects, making it necessary to work with whatever dimensional parameters are available.

Most DNN models achieved their best results in iteration 6, suggesting that further iterations with more neurons could yield even better performance.

Acknowledgements

This paper is partly funded by the Smart European Shipbuilding Projekt (SEUS).

References

ABRIL-PLA, O.; ANDREANI, V.; CARROLL, C.; DONG, L.; FONNESBECK, C.J.; KOCHUROV, M.; KUMAR, R.; LAO, J.; LUHMANN, C.C.; MARTIN, O.A.; et al. (2023), *PyMC: a modern, and comprehensive probabilistic programming framework in Python*, PeerJ. Computer Science 9, p. e1516, https://doi.org/10.7717/peerj-cs.1516

ALKAN, A.D.; GULEZ, K.; YILMAZ, H. (2004), *Design of a robust neural network structure for determining initial stability particulars of fishing vessels*, Ocean Eng. 31(5–6), pp.761–777

AUSTRALIA, R.; WILSON MININGA, W. (n.d.), *Commission of Inquiry into the sinking of Rabaul Queen COMMISSION OF INQUIRY INTO THE SINKING OF RABAUL QUEEN*, https://www.academia.edu/29381006/Commission_of_Inquiry_into_the_sinking_of_Rabaul_Queen_COMMISSION_OF_INQUIRY_INTO_THE_SINKING_OF_RABAUL_QUEEN

BAGAZINSKI, N.J.; AHMED, F. (2023), *Ship-D: Ship Hull Dataset for Design Optimization using Machine Learning*, arXiv, https://doi.org/10.48550/ARXIV.2305.08279

BEGOVIC, E.; BERTORELLO, C.; RINAURO, B.; ROSANO, G. (2023), *Simplified operational guidance for second generation intact stability criteria*, Ocean Eng. 270, p. 113583

BIERKOWSKA, K.; HINZ, T.; MAZERSKI, G. (2025), *Application of Neural Networks in Early-Stage Ship Design for Stability Evaluation Using IMO Second*, Innovations in Sustainable Maritime Technology (IMAM 2025)

BLEI, D.M.; KUCUKELBIR, A.; McAULIFFE, J.D. (2017), *Variational Inference: A Review for Statisticians*, J. American Statistical Association 112(518), pp.859–877

CHICCO, D.; JURMAN, G. (2020), *The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation*, BMC Genomics 21(1), p.6, https://doi.org/10.1186/s12864-019-6413-7

FRANCE, W.N.; LEVADOU, M.; TREAKLE, T.W.; PAULLING, J.R.; MICHEL, R.K.; MOORE, C. (2003), *An Investigation of Head-Sea Parametric Rolling and Its Influence on Container Lashing Systems*, Marine Technology and SNAME News, 40(01), pp.1–19

FRIEDMAN, J.H. (2001), *Greedy function approximation: A gradient boosting machine*, The Annals of Statistics 29(5), https://doi.org/10.1214/aos/1013203451

HARRIES, S.; ABT, C.; BRENNER, M. (2019), *Upfront CAD—Parametric Modeling Techniques for Shape Optimization*, Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences, Springer pp.191–211

ICHINOSE, Y.; GASPAR, H.M. (2023), *Interactive Ship Flow Simulation Enhanced by Neural Network Model in a Web Environment*, ECMS, pp.155–161, https://doi.org/10.7148/2023-0155

IMO (2009), *International Code on Intact Stability, 2008*, Int. Mar. Org., London

IMO (2020), *Interim guidelines on the second generation intact stability criteria*, Int. Mar. Org., London

IMO (2023), *Explanatory Notes to the Interim Guidelines on the Second Generation Intact Stability Criteria*. MSC.1/Circ.1652, Int. Mar. Org. London

IMO (2024), *Explanatory Notes to the Interim Guidelines on the Second Generation Intact Stability Criteria*. MSC.1/Circ.1652/Corr.1, Int. Mar. Org. London

KAUFMANN, J. (2009), *Fatal accident on board the CMV Chicago express during typhoon "Hugupit" on September 24 2008 off the Coast of Hong Kong*, Investigation Report 510(08), Bundes-stelle für Seeunfalluntersuchung,

KHAN, S.; GOUCHER-LAMBERT, K.; KOSTAS, K.; KAKLIS, P. (2023), *ShipHullGAN: A generic parametric modeller for ship hull design using deep convolutional generative model*, Computer Methods in Applied Mechanics and Eng. 411, p.116051, https://doi.org/10.1016/j.cma.2023.116051

KINGMA, D.P.; BA, J. (2014), *Adam: A Method for Stochastic Optimization*, arXiv, https://arxiv.org/pdf/1412.6980

NESS, R.O. (2025), *Causal AI*, Manning

PANDA, J.P. (2023), *Machine learning for naval architecture, ocean and marine engineering*, J. Marine Science and Technology 28(1), pp.1–26

PEDREGOSA, F.; VAROQUAUX, G; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; et al. (2011), *Scikit-Iearn: Machine learning in python*, J. Machine Learning Research 12

PETACCO, N.; GUALENI, P. (2020), *IMO Second Generation Intact Stability Criteria: General Overview and Focus on Operational Measures*, J. Marine Science and Engineering 8(7), p.494, https://doi.org/10.3390/jmse8070494

PINHEIRO CINELLI, L.; ARAÚJO MARINS, M.; ANTÚNIO BARROS DA SILVA, E.; LIMA NETTO, S. (2021), *Bayesian Neural Networks*, Variational Methods for Machine Learning with Applications to Deep Networks, Springer, pp.65–109

SALEHIN, I.; KANG, D.K. (2023), *A Review on Dropout Regularization Approaches for Deep Neural Networks within the Scholarly Domain*, Electronics 12(14), p.3106, https://doi.org/10.3390/electronics12143106

SAYLI, A.; ALKAN, A.D.; NABERGOJ, R.; UYSAL, A.O. (2007), *Seakeeping assessment of fishing vessels in conceptual design stage*, Ocean Eng. 34(5–6), pp.724–738

SHANMUGAMANI, R. (2018), *Deep Learning for Computer Vision: Expert Techniques to Train Advanced Neural Networks Using TensorFlow and Keras*, Packt Publishing

VASILEV, M.; KALAJDŽIĆ, M.; IVKOVIĆ, I. (2024), *CFD-Powered Ship Trim Optimization: Integrating ANN for User-Friendly Software Tool Development*, J. Marine Science and Eng. 12(8), p.1265, https://doi.org/10.3390/jmse12081265

WATT, J.; BORHANI, R.; KATSAGGELOS, A.K. (2020), *Machine Learning Refined: Foundations, Algorithms, and Applications*, Cambridge University Press

YALE, K.; NISBET, R.; MINER, G.D. (2017), *Handbook of Statistical Analysis and Data Mining Applications*, Elsevier

Ships in Sea-Ice – A Survey of Simulation Technology

Thomas DeNucci, USCG Academy, New London/USA, thomas.w.denucci@uscga.edu
Hazel Mitrik, USCG Academy, New London/USA, hazel.a.mitrik@uscga.edu
Volker Bertram, DNV, Hamburg/Germany, volker.bertram@dnv.com

Abstract

This paper surveys numerical approaches to simulate ships and offshore structures in ice-infested waters, excluding the direct simulation of icebreaking. Discrete element techniques model many individual ice floes, while classical continuum mechanics and computation fluid dynamics model the flow of water. Machine learning based simulations also show promise in predicting ship behavior in ice conditions. The state of the art has reached a maturity sufficient for many industrial applications.

1. Introduction

The Arctic is undergoing a rapid transformation due to climate change, with significant reductions in sea ice extent increasingly observed in recent decades. According to the *National Snow and Ice Data Centre (2021)*, Arctic Sea ice extent has declined by approximately 40% since the late 1970s, leading to improved accessibility for maritime activities. As the ice cover decreases, shipping activity in these previously hard-to-navigate waters has surged, driven by a combination of geopolitics, commercial interests, resource exploration, and new shipping routes. The Northern Sea Route and the Northwest Passage are now increasingly viable alternatives for global shipping, offering substantial reductions in transit times and fuel costs compared to traditional routes through the Suez or Panama Canals.

Recent statistics underscore this trend. According to the *Arctic Council (2024)*, the number of unique ships entering the Arctic region has risen by 37% from 2013 to 2024. In 2024 alone, 1,781 unique ships entered the arctic waters—nearly 500 more than when data collection began in 2013. This increase reflects a broader shift, as year-round shipping operations and seasonal voyages are establishing themselves as standard practices in the region. As a result, there is a need for improved understanding of ship-ice interactions in these changing sea conditions.

2. Ice Types

Sea ice in polar regions can be broadly categorized into two main types: level ice, which exists as a continuous surface, and broken ice, which consists of discontinuous pieces. According to *Li and Huang (2022)*, broken ice includes various forms such as brash ice, sliding ice pieces, and ice floe fields that emerge through natural processes. Broken ice features irregularly shaped fragments of varying sizes, ranging from small chunks to larger plates, with thicknesses that can significantly differ. These variations in size and thickness influence the stability and buoyancy of the ice.

Brash ice is defined as an accumulation of floating ice fragments that are no larger than 2 m across. It is characterized by its relatively small size and typically forms an ice layer over broken channels at full concentration. In the context of channels, brash ice is specifically characterized as being thicker than one layer. Brash ice also does not include features such as rafted ice or pressure ridges; it strictly pertains to these floating fragments, a characteristic that plays a crucial role in understanding the ship-ice dynamics in shipping channels, *Griesman (1981).*

Ice floe fields, on the other hand, are defined by numerous characteristics including ice concentration, diameter, thickness, and shape. An ice floe is a relatively flat piece of sea ice that measures 10 m or more across. Floes are categorized based on their horizontal extent: small (10-100 m), medium (100-500 m), big (500-2000 m), vast (2-10 km), and giant (greater than 10 km). These floes can freeze together to form larger ice fields and are distinct in that they float on the sea surface. Fig.1 shows U.S. Coast Guard ships operating in brash ice and in ice floes.

Fig.1: Vessels operating in brash ice (left) and in ice floes (right)

3. Ice Resistance Prediction Methods

The primary methods employed for predicting ice resistance include full-scale tests, ice tank model tests, empirical formulas, and numerical simulations. Among these methods, full-scale testing is widely regarded as the most reliable approach. However, it is not suitable for concept design and is usually conducted at trials and data gathered helps inform other models. Model tests conducted in ice tanks offer the most direct means of estimating ice resistance. *Mueller and Ettema (1984)* conducted model experiments with a scaled version of the USCGC Polar Star to investigate the interaction between ice forces and the dynamic motion of the hull as it traversed an ice sheet. *Daley (1991)* investigated and proposed the contact process at the edge of the ice layer using real ship tests and model test methods. *Matala and Suominen (2023)* developed a new scaling principle for model tests in old brash ice channels by considering the interparticle cohesion of the ice. The channel similitude number scaled-down flexural strength altered the ice fragment interaction, which is a substantial factor in ship's resistance in an old brash ice channel and results in resistances lower than those determined using current scaling principles.

Table I: Summary of different ice resistance prediction methods and limitations, *Sun et al. (2024)*

Method	Test Conditions	Computational Cost	Suitability	Research Gap
Full-scale test	Ice-covered water	-	Accurate and reliable data	Expensive and non-replicable
Model test	Model ships and ice water pools etc.	-	Applicable to existing or newly designed model ships	Expensive and time-consuming testing
Empirical/ semi-empirical formula	Simplify ship-ice interaction through reasonable assumptions, combining tests	Low	Application background, assumed conditions, scope of application	Should improve the applicability and prediction accuracy
Numerical model	Deeply understand the mechanism of ship ice interaction and proficiently apply numerical software	High	Compared with experimental methods, low cost and strong replicability	Often simplifying and assuming some conditions
Deep learning	Dataset collection, data preprocessing, and model selection optimization, etc.	Training: medium Calculation: low	Theoretically applicable to all scenarios of ships	Scale effect, high requirements for dataset quality

While the advancements in model testing have significantly enhanced our understanding of ice resistance, it is important to acknowledge the inherent limitations that accompany these methodologies. Conducting model tests can be time-consuming and costly, particularly for comprehensive testing across different scenarios, which might limit the number of tests that can be performed. Model tests

often focus on certain predefined scenarios, and they may not address all aspects of vessel maneuvering in ice, such as dynamic response to changing conditions or emergency maneuvers. Table I summarizes the different ice prediction methods along with their required test conditions, computational cost and suitability.

4. Semi-empirical and qualitative approaches

Over the years, a variety of ice resistance formulas have been developed. *Lindqvist (1989)* developed a relatively straightforward empirical model that accounts for key ship hull characteristics, ice thickness, ice strength, and friction. This method paved the way for a standardized approach to assessing vessel resistance in level ice, with subsequent research aimed at refining its details, *Kämäräinen (2007). Riska and Leiviskä (1997)* developed an approach for calculating resistance in level ice, utilizing empirical coefficients derived from full-scale data collected from several ships operating in the Baltic Sea. In one of the most comprehensive studies on ship resistance to date, *Keinonen (1991, 1996)* analyzed data from 18 icebreakers through extensive fieldwork and operator interviews. *Spencer and Jones (2001)* explored ice resistance prediction methods, introducing a component-based approach that identifies four distinct resistance forces encountered during icebreaking. However, it is important to note that the accuracy of these empirical formulas can be significantly off, with calculation errors reaching up to 40% when compared to model test results.

Recent improvements in computing power have redirected much of the research effort toward the reformulation of semi-empirical equations used to measure ice resistance. Most approaches differentiate ice resistance into two primary components: breaking (or crushing) and submersion (or bending). A detailed literature review is presented in *Kämäräinen (1993)*, which features several validation cases. More recently, *Erceg and Ehlers (2017)* examined six semi-empirical methods for estimating level ice resistance, testing them on four vessels of differing sizes and bow designs. Their results are supported by full-scale resistance measurements, offering valuable insights into the effectiveness of these methods. Researchers are actively working to enhance existing ice prediction techniques while also developing new ones. Examples of this include *Cho et al. (2014), Jeong et al. (2017),* and *Myland and Ehlers (2019)*, who have integrated traditional semi-empirical approaches into numerical models.

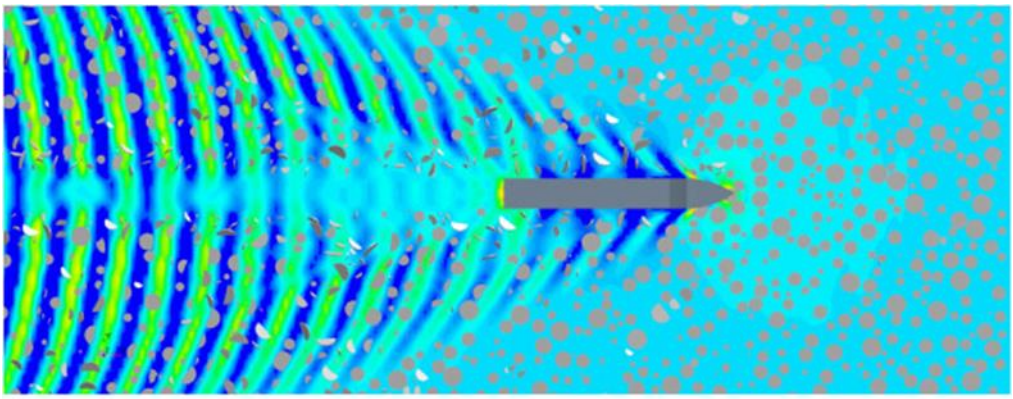
5. Introduction to numerical simulation of ice

Numerical simulations are fundamentally more precise than semi-empirical methods, as they are grounded in first principles; however, they can also be computationally intensive. Several approaches exist for modeling ice in numerical simulations:

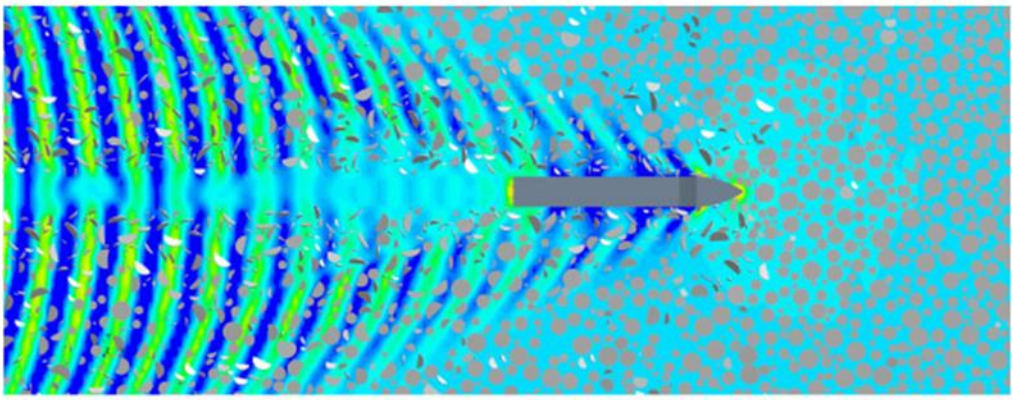
- Discrete Element Modelling (DEM): This technique involves modeling and tracking individual ice floes with discrete element models, widely adopted for simulating ice-structure interactions, *Tuhkuri and Polojärvi (2018)*. Typically, ice floe shapes are predefined, with circular and rectangular forms being the most used. Other studies, including those by *Yamaguchi et al. (1997)* and *Konno and Mizuki (2006)*, explore shapes like paralepidids, while *Wang and Derradji-Aounat (2011)* examine more arbitrary forms.
- Finite Element Method (FEM): FEM simulates ice behavior by subdividing the ice domain into finite elements. Its strengths lie in effectively modeling contact collisions, deformation, and damage, as well as the interactions between the hull and the ice, *Kuutti et al., (2013)*. For instance, *Kim et al. (2019)* employed FEM to analyze the loads exerted on a ship's hull due to ice impacts in fragmented ice fields. They modeled interactions using parameters such as drag coefficients and pressure-penetration curves, derived from comprehensive analyses with the coupled Eulerian-Lagrangian method.
- Computational Fluid Dynamics (CFD): Often used in conjunction with other models, CFD simulates interactions among ice, water, and ships. *Huang et al. (2020)* integrated CFD and DEM to explore the dynamics between ships, waves, and ice, Fig.2. *Zhong et al. (2023)*

developed both experimental and numerical methods to study ice floe-induced hull pressure, facilitating localized monitoring of these pressures within the CFD-DEM framework.

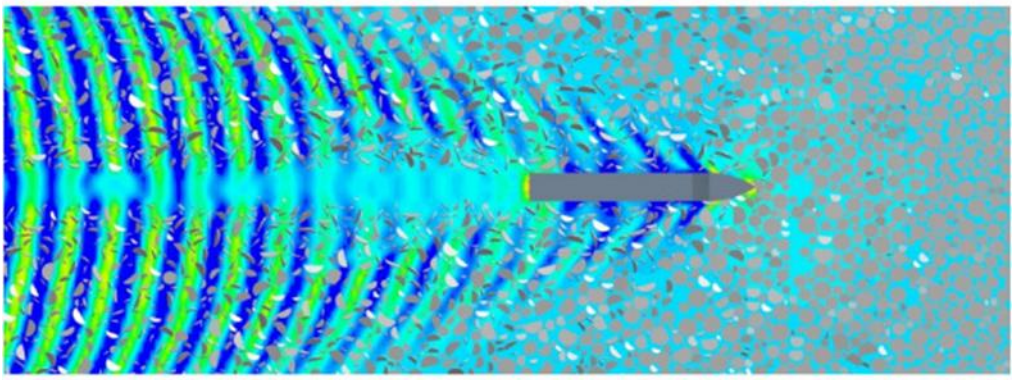
- Machine Learning and Data-Driven Techniques: Recent advancements have introduced machine learning models that can predict ice behavior using historical data, environmental factors, and observed patterns from simulations or real-world situations. *Sun et al. (2024)* proposed a deep learning method for generating data related to ice behavior.



(a) Ice concentration = 30%



(b) Ice concentration = 50%



(c) Ice concentration = 70%

Fig.2: Simulation of a ship advancing in floating ice floes, *Huang et al. (2020)*

Over the past decade, numerical simulations for ships operating in ice have seen considerable progress, yet the field remains relatively underdeveloped. The current landscape consists primarily of specialized research communities focused on innovation rather than commercial product

development, *Veeramani et al. (2020)*. This domain features a wide variety of methodologies, often varying in origin - from internal development projects to established commercial software solutions, *Thomas and Kuhl (2019)*. Many approaches tend to concentrate on specific aspects of ship-ice interactions, highlighting that the field is still evolving. Consequently, the current state of the art appears somewhat immature, emphasizing research efforts aimed at refining and innovating techniques rather than delivering robust commercial applications, *Jiang et al. (2021)*.

6. Numerical simulation of brash ice

Empirical estimation of broken ice resistance is relatively uncommon compared to established formulae for level ice resistance. *Riska et al. (1997)* present formulae specifically for brash ice, channel ice, and ice ridges. However, numerical simulation has emerged as a powerful tool to model ship-ice interactions with enhanced speed, fidelity, and accuracy. Given the discrete nature of broken ice, DEM techniques are particularly useful in modelling broken ice.

6.1. Discrete Element Modeling (DEM)

DEM offers significant advantages for calculating ice loads due to its ability to represent the discrete nature of ice at the micro-scale while accurately modeling the icebreaking phase during ship–ice interactions at a macro level, *Xue et al. (2020)*. Since the foundational work by *Cundall and Strack (1979)* on DEM for granular assemblies, the method has evolved and found numerous applications in ice simulation.

One of the early applications of DEM in ice modeling was by *Løset (1994)*, who developed a two-dimensional discrete element model to simulate dynamics and interaction forces within a broken ice field. This laid the groundwork for further studies. *Karulin and Karulina (2014)* utilized a two-dimensional disc DEM to explore the interaction processes between sea ice and ships operating at various speeds. Their findings were corroborated by comparing the results with existing ship model tests. *Prasanna and Hisette (2018)* focused on simulating ships navigating through brash ice. Their study demonstrated that while the particle behavior near the hull was satisfactory, the predicted ice loads tended to be higher than anticipated. Nevertheless, this tool proved valuable in providing qualitative insights into ship navigation within brash ice channels during the early design stage, particularly in visualizing ice particle flow around the hull and identifying potential concentrations of ice particles around appendages.

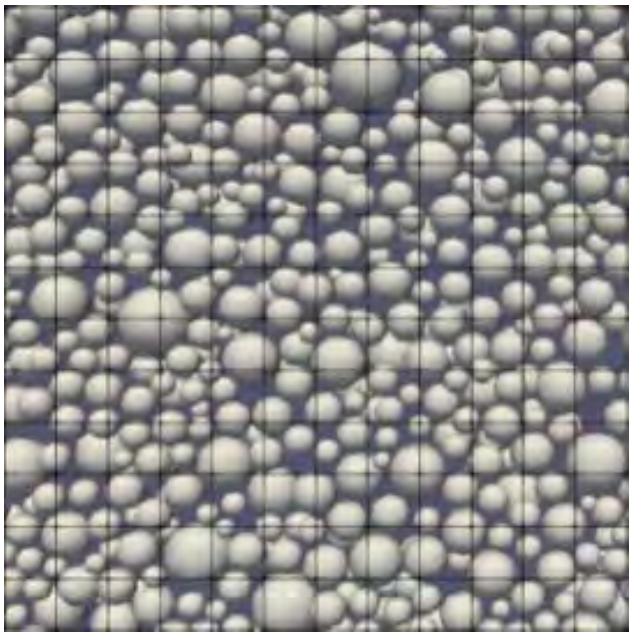


Fig.3: Detail of brash ice in the simulation, *Behen et. al (2020)*

In a more recent development, *Behen et al. (2020)* implemented a DEM model based on the Hertz-Mindlin Deresiewicz contact model, which accounts for elastic viscous behavior. Their simulation results showed significant consistency with HSVA data, though they also revealed a greater level of variation and larger individual peaks in the graphical representations, illustrated in Fig.3.

6.2. CFD-DEM Approaches

DEM is widely employed to study ice–ship interactions; however, it struggles to incorporate the effects of water alone. To address this, a hybrid approach combining CFD with DEM has gained traction. This method employs the Euler method to solve fluid flow while utilizing the Lagrangian method to model ice particle movement.

Vroegrik (2015) utilized the commercial code StarCCM+ with DEM to model brash ice flow around an obstacle. This study showcased the feasibility of CFD in tracking particle flow within environmental contexts, paving the way for further investigations that adopted hybrid CFD-DEM methodologies. *Luo et al. (2020)* implemented a CFD-DEM approach to simulate a bulk carrier navigating through brash ice. Their research involved coupling numerical methods to assess the ship resistance of an ice-strengthened Panama bulk carrier in these challenging conditions. They adopted advanced computational techniques, including finite element analysis, to precisely depict the ice's physical properties and the vessel's structural responses. Their results indicated a close correlation with experimental data from the HSVA ice tank, with an average error of approximately 5.13%.

Zhang et al. (2021) used CFD-DEM simulations to create a ship-ice contact model and compared it with full-scale ship-ice interaction data in a confined brash ice channel. The collision, accumulation, extrusion, rolling, and sliding behaviors observed in the simulations closely matched those seen in ice tank tests, Fig.4. This suggests that the proposed full-scale CFD-DEM method for directly assessing ship-brash ice interactions provides reliable results and holds practical significance. *Xie et al. (2023)* expanded the application of the CFD-DEM methodology to examine ship-ship interactions within brash ice, particularly noting the significant impact of proximity on these dynamics. Their approach simulated the hydrodynamic interactions and ice dynamics of multiple vessels navigating in close formations, utilizing specific rheological models to characterize the brash ice's properties and its interactions with ship hulls. The validity of their models was reinforced through comparisons with prior experimental and field data. *Zhang et al. (2024)* introduced wave effects into the coupled CFD-DEM analysis, employing overlapping grid technology to model ships' motion responses under the combined influences of nonlinear waves, currents, and brash ice. Their work examined both single vessels and formations navigating through these complex conditions, further enhancing the understanding of ice-ship interactions in maritime environments.

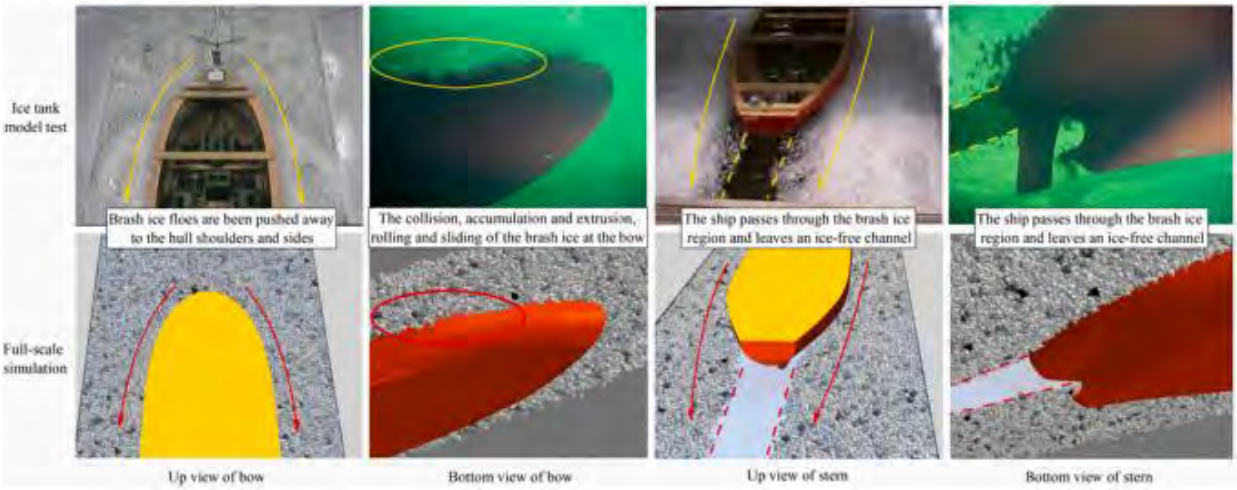


Fig.4: The brash ice distribution comparison between the ice tank model test and the corresponding full-scale simulation, *Zhang et al. (2021)*

Liu et al. (2025) utilized a CFD-DEM model incorporating an overset grid technique to investigate forward ship motion in brash ice channels. This research improved the understanding of ship-ice-water interactions by adopting a quasi-physical framework where the ship moves while the ice remains stationary. They emphasized the need for careful selection and calibration of contact model parameters, as these significantly influence the accuracy of ice resistance predictions.

Zou et al. (2024) utilized a hybrid CFD-DEM approach to simulate the interaction between the ship and the brash ice, allowing for a detailed investigation of the forces exerted on the ship hull, Fig.5. The ship's geometry and operational conditions were defined, and the simulation incorporated varying ice concentrations and channel conditions to evaluate how these factors influence the ice forces acting on the ship. The findings indicate that the distribution of ice forces on the ship hull is significantly affected by the ice concentration and the flow dynamics within the brash ice channel. The study revealed that as the ice concentration increases, the forces exerted on the ship's hull also rise, showcasing a direct correlation between these two variables.

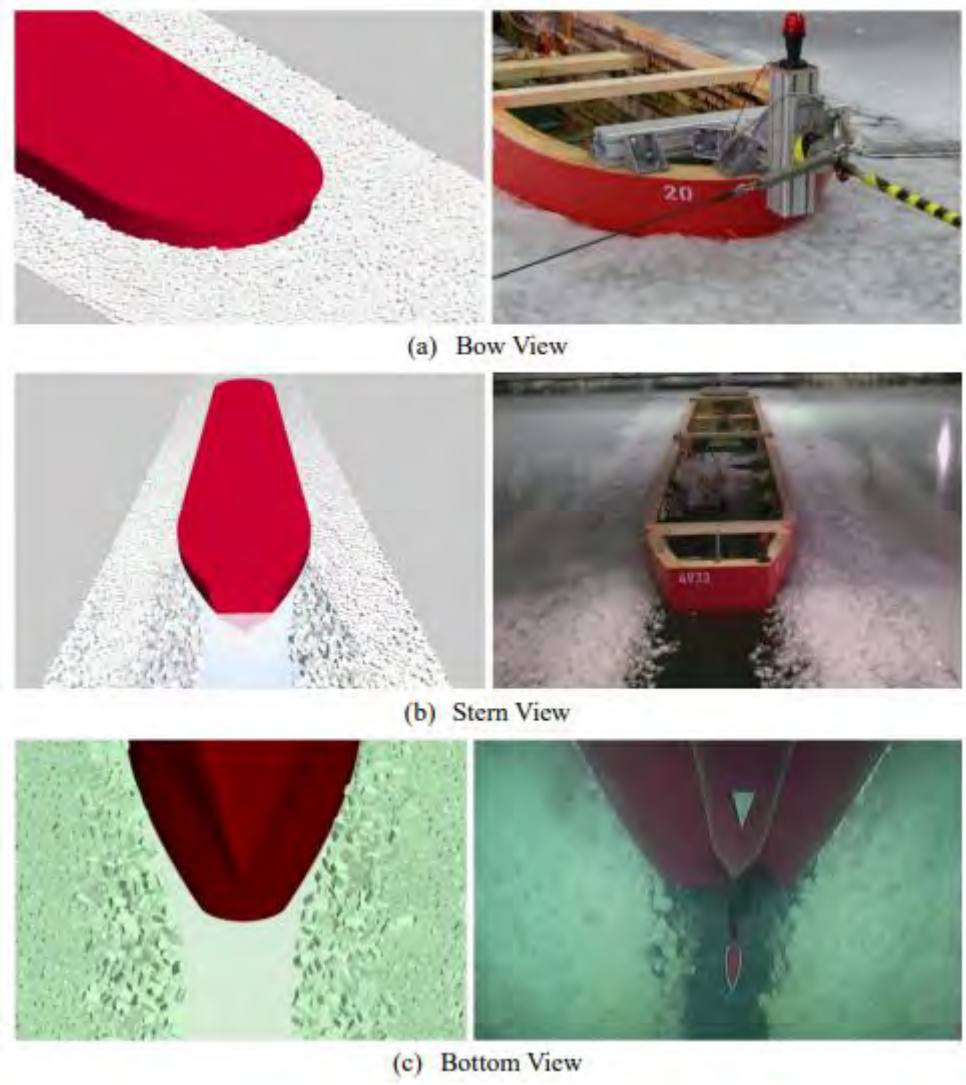


Fig.5: Simulations (left) and experiments (right), *Zou et al. (2024)*

6.3. Other numerical methods for brash ice

Konno et al. (2007) developed a physically based model to simulate a ship in brash ice with an open dynamics engine (ODE). Since the code was multipurpose and designed to deal with different scales of problems, it didn’t accurately capture friction and damping forces. Improvements to this method

were demonstrated in the model by *Konno and Yoshimoto (2008)*, which simulated a ship navigating through a channel filled with 1250 pieces of brash ice. *Konno (2009)* and *Watanabe and Konno (2011)* extended the approach to 33000 and 126000 pieces, respectively. They also integrated base water flow using the open-source CFD solver OpenFOAM.

Wang and Derradji-Aouat (2011) described a simulation involving ice floes drifting against an offshore structure, utilizing the commercial code LS-DYNA (www.lstc.com). In this simulation, ice floes were modeled as both rectangular and arbitrary shapes, revealing that shape had little impact on the overall results. The study explored three different ice concentrations, modeling water and air with the Eulerian method while employing the Lagrangian method for the offshore structure and pack ice. Semi-empirical friction coefficients were utilized to characterize the friction between ice floes and the structure. Figs.6 and 7 show rectangular and randomly shaped ice at varying concentrations.

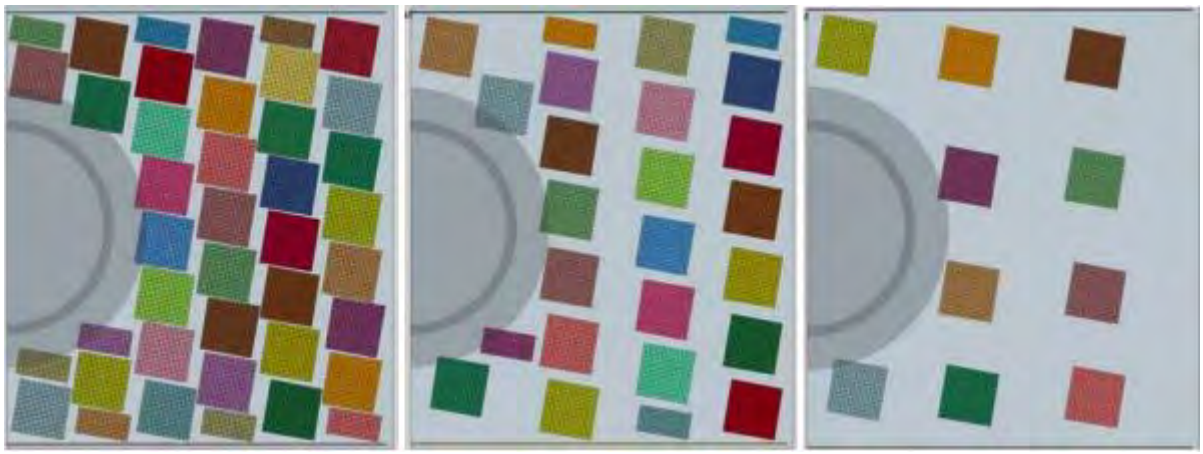


Fig.6: 80%, 50%, and 20% concentrations, *Wang and Derradji-Aouat (2011)*

Fig.7: 80%, 50%, and 20% concentrations with random shaped ice, *Wang and Derradji-Aouat (2011)*

Cabrera (2017) employed Smoothed Particle Hydrodynamics (SPH) to create a mesh-free simulation. This innovative model allowed for complex particle interactions through force calculations that included both viscous and pressure forces, yielding realistic representations of ice dynamics. Enhancing this technique, *Chen (2022)* tackled issues related to mesh distortions and disappearance in Finite Element Method (FEM) by integrating it with SPH. However, challenges remain, as simulating a ship using SPH in an enlarged water domain necessitates further development of existing code.

7. Numerical simulation of ice floes

Ice floes are the most prevalent type of fragmented ice found in polar regions. They can naturally form from newly frozen water or result from the breakup of larger ice masses due to heat or ocean

waves, *Li and Huang, (2022)*. The focus of numerical simulations involving ice floes is primarily on resistance, manoeuvrability, and the impacts of ice on the hull, rather than the specific formation methods. *Huang (2020)* addresses the challenges posed by operating ships in environments with floating ice floes by examining the various types of resistance that vessels encounter due to ice interaction and emphasizes the importance of understanding these factors for safe navigation and optimization of ship hulls for Arctic operations.

7.1. Ice floe resistance

Recent advancements in numerical simulation methods have significantly enhanced the understanding and prediction of ship resistance while navigating through pack ice. *Huang et al. (2024)* introduced a novel two-dimensional (2D) numerical simulation method designed specifically to predict the resistance faced by ships operating in pack ice. This approach is characterized by a semi-theoretical and empirical framework. Comparative analyses revealed that the simulations effectively reproduced the interactions between the ships and ice, successfully capturing critical ice failure modes such as sliding, splitting, and bending. Notably, the total discrepancy between the calculated average navigation resistance from the simulations and the resistance measured during the model tests was a mere 9.05%. *Huang et al. (2024)* also explored the modeling of smaller ice floes. They employed a coupled CFD–DEM simulation to assess ice resistance in smaller floes, considering the effects of ship-generated waves. This aspect is crucial for enhancing the accuracy of predictions in varying ice conditions.

The application of advanced computational methods has further refined predictions of ship resistance in ice. *Huang et al. (2022)* highlighted the widespread use of the Finite Element Method (FEM) and Discrete Element Method (DEM) in this domain. These methodologies are often integrated with commercial Computational Fluid Dynamics (CFD) software, enabling the development of combined CFD–FEM and CFD–DEM simulation approaches. This integration has proven effective in simulating ship resistance in ice floes, as suggested by *Zhong (2023)*.

7.2. Manoeuvring

Zhan and Molyneux (2010) utilize the 2D version of DECICE (Discrete Element Code for Ice-related problems) to analyse tug-tanker ship manoeuvring in ice. *Lau and Simoes Ré (2006)* used the code to model a ship operating in pack ice. *Lawrence (2009)* expanded the code's capabilities in his PhD thesis. *Liu et al. (2010)* applied DECICE3D—a 3D version of DECICE—to simulate offshore structures, including a drill ship, navigating through pack ice. In a subsequent study, Zhan and *Molyneux (2012)* also employed DECICE3D for ship manoeuvring in icy conditions. DECICE functions as a time-domain solver, employing discrete elements to tackle ice engineering challenges. These elements experience rigid-body translation and rotation based on classical Newtonian mechanics. Interaction forces, including both normal and tangential forces arising from friction, are modelled during contacts between bodies (whether floe-floe or floe-ship). *Molyneux et al. (2012)* validated DECICE simulations against model tests, and their findings for three different ship designs and varying ice conditions demonstrated a strong correlation with the results from experimental models.

Wang et al. (2020) used a finite element to simulate the navigation of an ice-area bulk carrier in ice floes. The ice is defined as elastic, and the simulations, Fig.8, were accomplished at four model speeds and three ice concentrations. In terms of ship-ice-water interaction phenomena, the ice floes tend to overlap, accumulate, and overturn at the bow and slip along the side of the ship. Most ice floes that accumulate and slip are small and medium-sized, and larger ones are more prone to overturning. *Yang et al. (2024)* coupled a non-smooth discrete element method (NDEM) with the Manoeuvring Modelling Group (MMG) model to simulate the manoeuvring motion of full-scale ships in broken ice. The simulation results demonstrated significant variations in maneuvering performance based on ice concentration and the physical characteristics of ice.

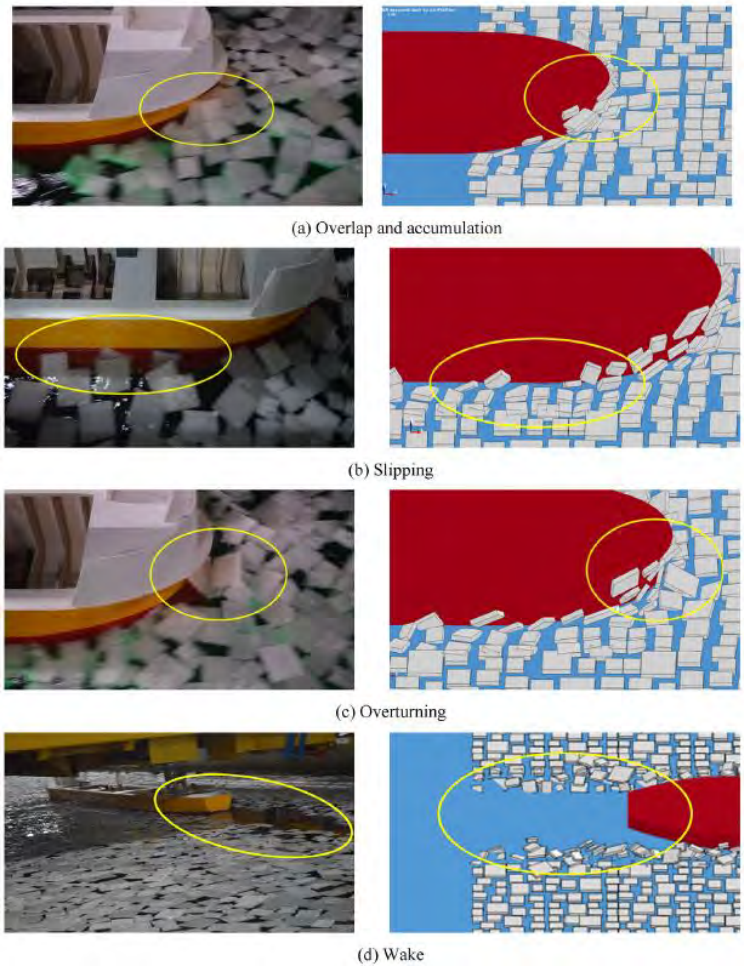


Fig.8. Simulation of various ship-ice interactions, *Wang et al. (2020)*

7.3. Ice impacts

Kim et al. (2019) conducted numerical simulations to analyze the impacts of ice on ship hulls navigating through broken ice fields. They utilized a CFD approach to model the interactions between ice and ship structures, incorporating various parameters such as ice thickness, ship speed, and hull design. The methodology involved creating a detailed simulation environment to replicate real-world conditions, allowing for the examination of stress distributions and potential damage to the hull during ice impacts. Results of the hull-ice interaction, Fig.9, indicated that the severity of ice impacts varied significantly based on these parameters, highlighting critical vulnerabilities in hull designs and providing insights for improving ship safety in icy waters.

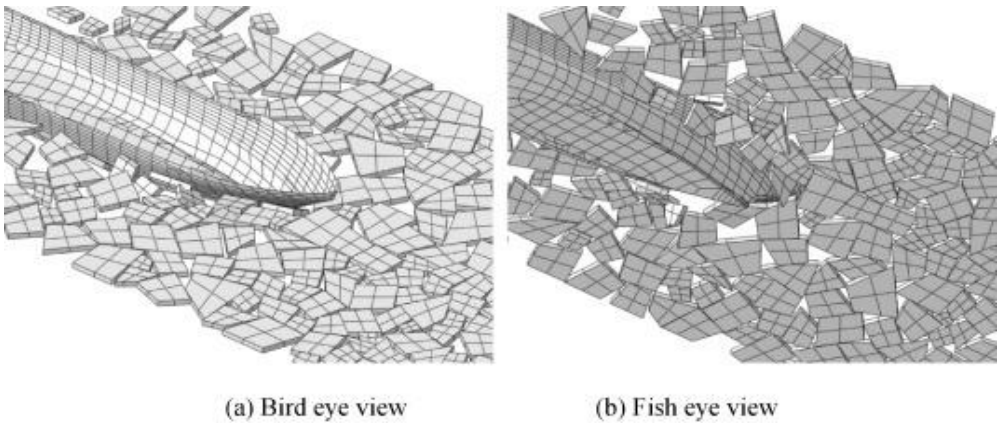


Fig.9: Interaction between ice floes and the ship hull at an 80% concentration, *Kim et al. (2019)*

Xu et al. (2024) conducted a numerical simulation to investigate ice-water-ship interactions using a novel adaptive coupling algorithm combining Finite Element Method (FEM) and Smoothed Particle Hydrodynamics (SPH). The methodology involved creating a detailed computational model that represented the physical interactions between the ship hull, ice, and water, with the adaptive algorithm allowing for effective handling of both solid and fluid mechanics. The results demonstrated that the proposed model accurately captured the complexities of ice breakage and water flow, revealing critical insights into the forces exerted on the ship during ice navigation.

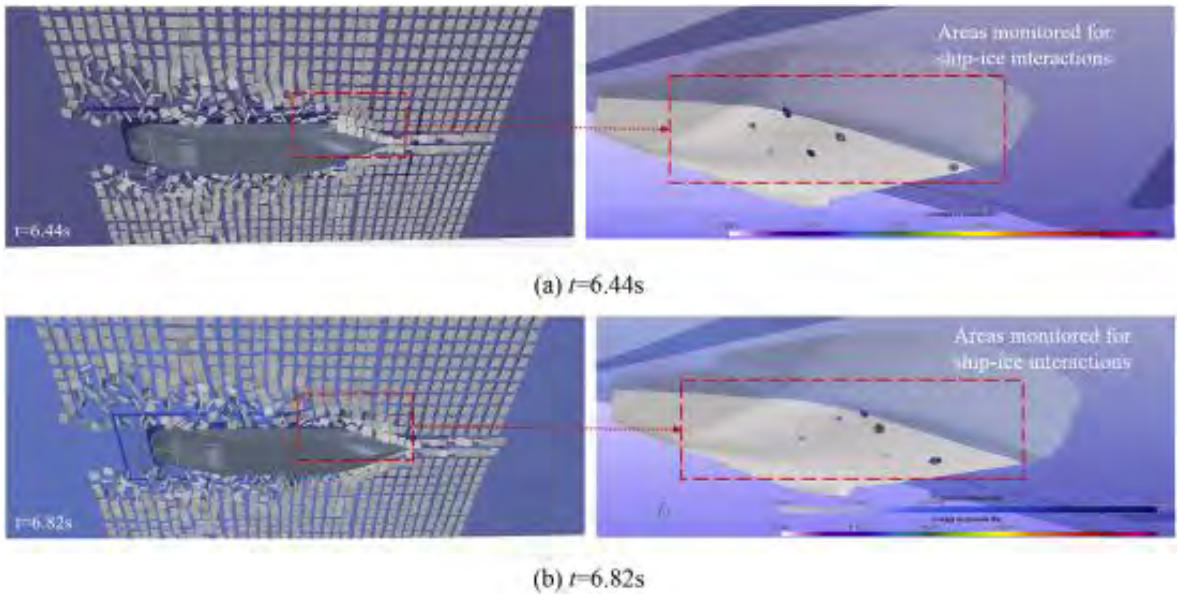


Fig.10: Simulated ship-ice interaction and pressure contour, *Zhong et al. (2023)*

Zhong et al. (2023) employed a CFD-DEM approach to investigate how ice floe shapes, sizes, and ship speeds affect hull pressure characteristics, including water–ice interactions at the hull. The study combined direct measurements with CFD simulations to assess ice-induced hull pressures experienced by ships navigating through floe-ice fields. The researchers conducted field experiments to gather real-time data on hull pressures while a ship operated in ice-laden waters, allowing for precise calibration of their CFD models. The results, Fig.10, demonstrated a strong correlation between measured pressures and simulated values, validating the effectiveness of the CFD approach. *Zhong et al. (2024)* analyzed ice-induced loads on ship hulls in regions with ice floes using a coupled computational fluid dynamics (CFD) and discrete element modeling (DEM) approach. The methodology involved simulating the interaction between ship hulls and ice floes to quantify the loads experienced by the hulls under various conditions. The results indicated that the ice-induced loads were significantly influenced by factors such as ice thickness, floe size, and ship speed.

Li et al. (2020) presents a methodology for modeling ship-ice interaction using finite element analysis (FEA) to enhance ship performance simulations, specifically focusing on the shoulder and midship areas of the vessel. The authors developed a meta-model to effectively capture the complex interactions between the ship hull and ice, incorporating various factors such as ice properties and ship operational parameters. The methodology involved creating a finite element model to run simulations, which were then used to train the meta-model for quicker predictions. Results indicate that the meta-model significantly improves the accuracy and efficiency of ship performance predictions in icy conditions compared to traditional methods.

8. Ice Simulations using Machine Learning (ML)

Artificial intelligence is a rapidly expanding industry with increasingly widespread applications. Deep learning is a more recent enhancement to the webs of artificial intelligence that has been expanding into areas of traditional knowledge and process-based modelling, *Razavi (2021)*. Deep learning

modeling relies on the availability and iteration of existing data as input features for the formation of its neural network for progressive processing, *Sun et al. (2024).*

Fu et al. (2016) combined data with expert knowledge using Bayesian belief networks to develop probabilistic model for estimating the probability of ship besetting in ice along the Northeast Passage. The researchers employed historical data on ice coverage, vessel characteristics, and environmental conditions to train the model, which was subsequently validated using case studies. The results indicate that the model successfully forecasts the likelihood of vessels becoming beset in ice, providing valuable insights for maritime navigation and safety in challenging Arctic environments.

Li et al. (2020) utilized FEM modeling combined with a neural network to study ship-ice interaction. The approach preserves the accuracy of traditional FEM at the ship-ice interaction scale which significantly reduces computational costs, making it suitable for ship scale simulations. The ice failure process was initially examined quantitatively through full-scale and model-scale observations, along with numerical simulations using the Extended Finite Element Method (XFEM). The model was then simplified and implemented in Abaqus to automatically generate a large database. A neural network was trained on these results to create a fast, simulation-free tool for predicting ship-ice interactions.

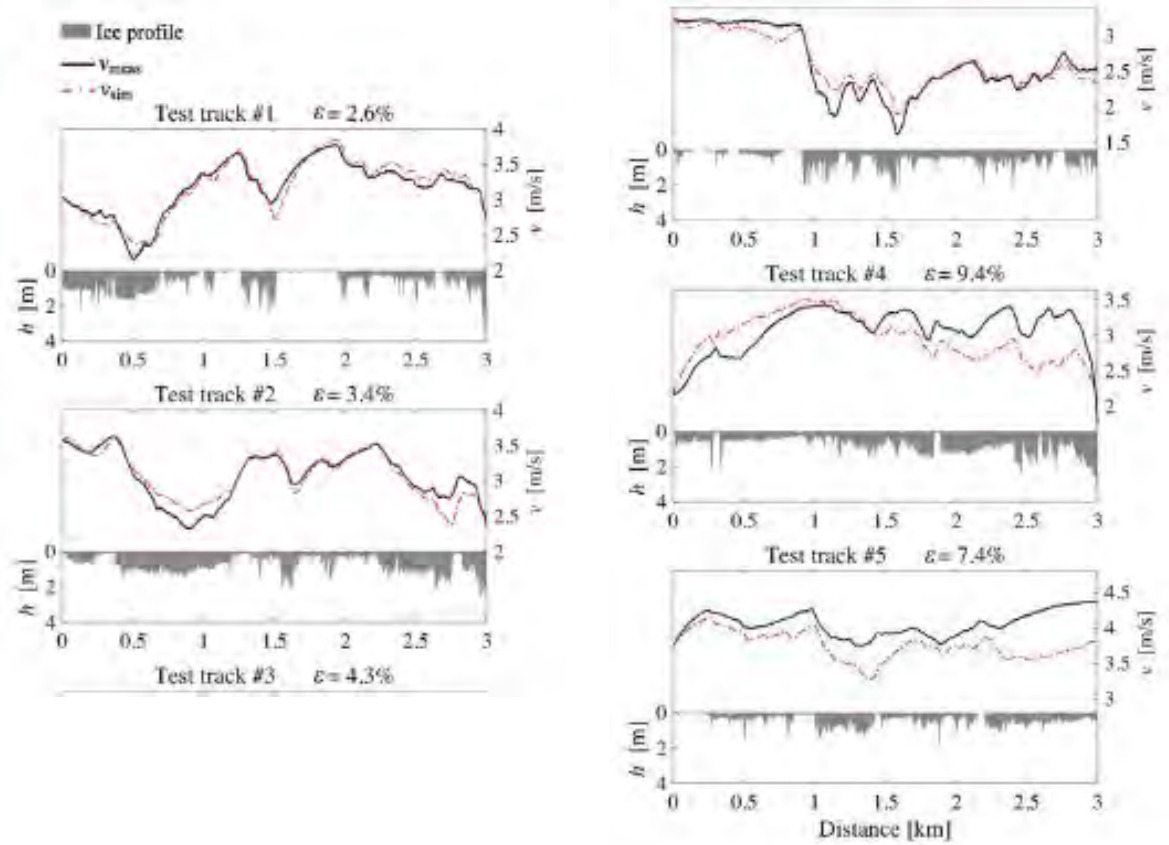


Fig.11: Test results of the ML-based simulator for each of the five test tracks, *Milaković et al. (2020)*

Milaković et al. (2020) introduced the first machine learning-based simulator for predicting ship speed profiles in complex ice conditions. The approach involved gathering extensive data on both ice characteristics and ship performance, then applying a range of machine learning algorithms to model the relationship between environmental conditions and ship speed. The simulator was trained and validated using data from full-scale trials of an icebreaking vessel. The model achieved strong accuracy, Fig.11, with average errors between the simulated and actual ship speeds ranging from 2.6% to 9.4%. These results demonstrate the capability of machine learning models to reliably estimate ship speed across varying ice conditions, underscoring their potential to enhance navigational safety and operational efficiency in Arctic shipping.

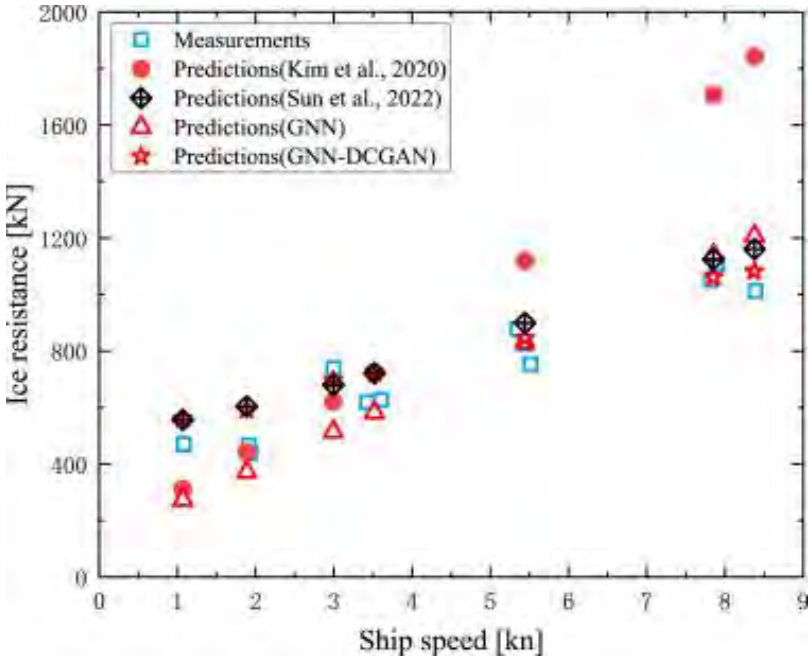


Fig.12: Three ANN methods predicting ice resistance based on Otso ship test data, *Sun et al. (2024)*

Sun et al. (2022) developed an Artificial Neural Network (ANN) model for the estimation of total ship resistance in ice-covered waters, with a focus on isolating and predicting the ice resistance component. The model was trained using a hybrid dataset composed of full-scale sea trial measurements and ice tank model test data. Validation results exhibited a high degree of correlation between predicted and empirical resistance values, indicating the ANN's capacity to generalize across varying ice conditions and vessel geometries.

Building on this, *Zhou et al. (2023)* presented a supervised machine learning framework for the prediction of propulsion power requirements in ice-infested environments. The model integrates multiple input features − specifically ice thickness, ambient temperature, vessel speed, and hull form characteristics − and is trained on historical operational datasets. When benchmarked against both model-scale and full-scale trial data, the model demonstrated a mean absolute percentage error (MAPE) of approximately 14%, indicating robust predictive performance suitable for practical deployment in voyage planning and energy management systems under polar operating conditions.

Zhong et al. (2024) addressed the problem of autonomous navigation in ice-covered regions by developing a machine learning-based predictive model for ship-ice interaction dynamics. The approach employs data-driven learning algorithms to infer spatial-temporal patterns of ice impact forces and resistance, facilitating enhanced path planning and real-time navigational decision-making. The model is designed to operate within autonomous control frameworks, contributing to increased situational awareness and operational safety in dynamic and unstructured polar environments.

Sun et al. (2024) advanced the state of the art in ice resistance prediction by developing a deep neural network model tailored for icebreaker hull design applications. The input feature space includes 11 parameters: principal ship dimensions (length, breadth, draft), hull form attributes (stem angle), kinematic variables (speed), and ice properties (thickness, flexural strength, etc.). The model was trained on a dataset of 300 entries derived from ice tank experiments covering 21 distinct hull types. Preprocessing included normalization and dimensionality reduction to address heterogeneity in input scales. A scale extrapolation factor was applied to enable generalization to full-scale operational scenarios. The model achieved high prediction accuracy, affirming its potential utility in early-stage design optimization and performance assessment under ice-loading conditions.

To enhance data generation and improve model robustness, the study also explored the use of Generative Adversarial Networks (GANs), addressing common challenges such as gradient vanishing and unstable outputs. A novel approach was proposed—combining Gaussian fuzzy preprocessing with a Deep Convolutional GAN (DCGAN) architecture, wherein the conventional feedforward neural network is replaced by a Graph Neural Network (GNN) to more effectively capture inter-feature relationships. The GNN-enhanced GAN architecture yielded improved predictive performance and stronger correlation with measured data compared to conventional methods, Fig.12. Nonetheless, the authors noted persistent challenges in GNN-based data optimization, emphasizing the critical role of high-quality data preprocessing and representation in the application of machine learning to complex marine engineering problems.

9. Conclusions

The simulation of vessel performance in ice-infested waters has seen significant advancements through both physics-based numerical methods and data-driven machine learning (ML) approaches. Traditional numerical models, grounded in physical principles of ice mechanics and hydrodynamics, have matured to offer reliable predictions of ice resistance, ship-ice interaction forces, and maneuvering behavior. However, these methods often require extensive computational resources, detailed ice field characterization, and can be limited by model assumptions and simplifications.

In parallel, machine learning has emerged as a powerful complementary tool, offering the ability to learn complex nonlinear relationships from empirical data with reduced computational overhead. Recent developments demonstrate the successful application of artificial neural networks, deep learning architectures, and other ML algorithms in predicting ice resistance, propulsion power, and ship-ice interaction dynamics with promising accuracy. These models have been trained on diverse datasets, including model (ice tank) experiments, full-scale trials, and historical records, enabling scalable and adaptive prediction capabilities.

Moreover, the integration of ML techniques into autonomous navigation systems and digital twins holds potential for real-time decision support, performance optimization, and risk mitigation in operations. However, challenges remain in terms of data availability, model generalization across varying ice conditions and hull forms, and the interpretability of ML models.

Future research should focus on hybrid modeling frameworks that combine the strengths of physics-based simulations with data-driven learning, supported by high-quality, standardized datasets. Such integrated approaches are expected to enhance the fidelity, efficiency, and operational relevance of vessel simulations in ice-covered waters, thereby contributing to safer and more efficient Arctic and Antarctic maritime operations.

References

ARCTIC COUNCIL (2024), *Arctic Shipping on the Rise: What Trends Can Tell Us*, https://arctic-council.org/news/arctic-shipping-what-trends-can-tell-us/

BEHNEN, J.; EHLERS, S.; HISETTE, Q.; VON BOCK, F.; POLACH, R. (2020), *A Discrete element model for brash ice simulations,* 39th Int. Conf. Ocean Offshore and Arctic Engineering

CABRERA, I. (2017), *Smoothed particle hydrodynamics modeling of brash ice*, Master Thesis, University of Rostock

CHEN, C.; SHI, W.; SHEN, Y.; CHEN, J.; ZHANG, A. (2022), *A multi-resolution SPH-FEM method for fluid–structure interactions*, Comput. Methods Appl. Mech. Eng. 401

CHO, S.; JEONG, S.; LEE, S.; KANG, K. (2014), *Development of a Prediction Formula for Ship Resistance in Level Ice*, 33rd Int. Conf. Ocean, Offshore and Arctic Eng. 10

CUNDALL, P.; STRACK, O. (1979), *A Discrete Numerical Model for Granular Assemblies*, Geotechnique 29, pp. 47-65

DALEY, C. (1991), *Ice Edge Contact - A Brittle Failure Process Model*, Acta Polytechnica Scandinavica, Mech. Eng. Series No. 100

ERCEG, S.; EHLERS, S. (2017), *Semi-empirical level ice resistance prediction methods*, Ship Technology Research 64/1, pp.1–14

FU, S.; ZHANG, D.; MONTEWKA, J.; YAN, X.; ZIO, E. (2016), *Towards a probabilistic model for predicting ship besetting in ice in Arctic waters*, Reliability Eng. and System Safety 155, pp.124–136

GREISMAN, P. (1981), *Brash Ice Behavior*, U.S. Coast Guard Research and Development Center: Report No. USCG-D-30-81

HSVA (2012), *Brash Ice Tests for a Panmax Bulker with Ice Class 1B,* Report, HSVA, Hamburg

HUANG, L.; TUHKURI, J.; IGREC, B.; LI, M.; STAGONAS, D.; TOFFO, A.; CARDIFF, P.; THOMAS, G. (2020), *Ship resistance when operating in floating ice floes: A combined CFD&DEM approach*, Marine Structures 74

HUANG, Y.; SUN, C.; SUN, J. (2024), *Two-dimensional numerical method for predicting the resistance of ships in pack ice: development and validation,* J. Marine Science and Eng. 12

JEONG, S.; KYUNGSIK, C.; KUK-JIN, K.; HA, J. (2017), *Prediction of ship resistance in level ice based on empirical approach*, Int. J. Naval Arch. and Ocean Eng. 9/6, pp. 613-623

JIANG, Y.; WANG, Y.; ZHANG, H. (2021), *Current challenges and future directions in ice navigation research*, Polar Science 27

KÄMÄRÄINEN, J. (1993), *Evaluation of ship ice resistance calculation methods*, Licenciate Thesis, Helsinki University of Technology

KÄMÄRÄINEN, J. (2007), *Theoretical investigation on the effect of fluid flow between the hull of a ship and ice floes on ice resistance in level ice*, PhD thesis, Helsinki University of Technology

KEINONEN, A.; BROWNE, R.; REVILL, C. (1991), *Icebreaker Design Synthesis – Phase 2 – Analysis of Contemporary Icebreaker Performance,* Report for Transportation Development Centre, Transport Canada, TP 10923E

KEINONEN, A.; BROWNE, R.; REVILL, C.; REYNOLDS, A. (1991), *Icebreaker Characteristic Synthesis*, Report for Transportation Development Centre, Transport Canada, TP 12812E

KIM, J.; KIM, Y; KIM, H.; JEONG, S. (2019), *Numerical simulation of ice impacts on ship hulls in broken ice fields*, Ocean Eng. 182, pp.211-221

KONNO, A.; MIZUKI, T. (2006), *Numerical Simulation of pre-sawn ice test of model icebreaker using physically based monitoring*, 18th IAHR Int. Symp. on Ice, Vol. 2, Sapporo, pp.17-23

KONNO, A.; YASUKI, N.; YAMAMOTO, K.; WAKO, D.; TAKIMOTO, T.; IZUMIYAMA, K. (2007), *On the numerical analysis of flow around ice piece moving near icebreaker hull. Third report: comparison of simulation results with experimental results under pre-sawn ice condition*, 22nd Int. Symp. Okhotsk Sea and Sea Ice, Mombetsu, pp.29–32

KONNO, A. (2009), *Numerical simulation of ship navigation in brash ice channels. Second report:*

effect of size and layer number of ice pieces, 24th Int. Symp. Okhotsk Sea and Sea Ice, Mombetsu, pp.104-107

KONNO, A.; SAITOH, O.; WATANABE, Y. (2011), *Numerical investigation of effect of channel condition against ship resistance in brash ice channels*, 21st Int. Conf. Port and Ocean Eng. under Arctic Conditions (POAC), Montreal

KRAULIN, E.; KARULINA, M. (2011), *Numerical and physical simulations of moored tanker behaviour*, Ships Offshore Structures 6, pp.179–184

KUUTTI, J.; KOLARI, K.; MARJAVAARA, P. (2013), *Simulation of ice crushing experiments with cohesive* surface methodology, Cold Regions Science and Technology 92, pp.17-28

LAU, M.; SIMÕES RÉ, A. (2006), *Performance of survival craft in ice environments*, 7th Int. Conf. Performance of Ships and Structures in Ice (ICETECH), Banff

LAWRENCE, K. (2009), *Load prediction for a moored conical drillship in level unbroken ice: a discrete element and experimental investigation*, PhD Thesis, University of Waterloo

LI, F.; KORGESAAR, M.; KUJALA, P. GOERLANDT, F. (2020), *Finite element based meta-modeling of ship-ice interaction at shoulder and midship areas for ship performance simulation*, Marine Structures 71

LI, F.; HUANG, L. (2022), *A Review of Computational Simulation Methods for a Ship Advancing in Broken Ice*, J. Mar. Sci. Eng. 10(2)

LINDQVIST, G. (1989), *A straightforward method for calculation of ice resistance of ships*, 10th Int. Conf. Port and Ocean Eng. under Arctic Conditions (POAC), pp.722–735

LIU, Z.; YIN, Y.; ZHANG, X.; SUN, T.; WANG, S.; QIAN, X. (2025), *Numerical simulation of ship straight-line navigation in a pre-generated brash ice channel*, Ocean Eng. 339

LØSET, S. (1994), *Discrete element modelling of a broken ice field - Part I: model development*, Cold Regions Science and Technology 22, pp.339-347

LUO, W.; JIANG, D.; WU, T.; GUO, C.; WANG, C.; DENG, R.; DAI, S. (2020), *Numerical simulation of an ice-strengthened bulk carrier in brash ice channel*, Ocean Eng. 196

MATALA, R.; SUOMINEN, M. (2023), *Scaling principles for model testing in old brash ice channel*, Cold Regions Science and Technology 210

MILAKOVIĆ, A.; FANG, L.; MOHAMED, M.; SÖREN, E. (2020), *A machine learning-based method for simulation of ship speed profile in a complex ice field*, Ships and Offshore Structures 15, pp. 974-980

MUELLER, A.; ETTEMA, R. (1984), *Dynamic Response of an Icebreaker Hull to Ice Breaking.* Final Report, Iowa Institute of Hydraulic Research, Iowa City

MYLAND, D.; EHLERS, S. (2019), *Investigation on semi-empirical coefficients and exponents of a resistance prediction method for ships sailing ahead in level ice*, Ships and Offshore Structures, 14, pp.161–170

NSIDC (2021), *Arctic Sea ice extent*, National Snow and Ice Data Center, Boulder, https://nsidc.org/arcticseaicenews

PRASANNA, M.; HISETTE, Q. (2018), *Discrete Element Simulation of Ships Navigating Through Brash Ice Channels*, 50th OTC Arctic Technology Conf., Houston

RISKA, K.; WILHELMSON, M.; ENGLUND, K.; LEIVISKÄ, E. (1997), *Performance of merchant vessels in ice in the Baltic*, Helsinki University of Technology

SPENCER, J.; JONES, J. (2001), *Model-scale/full-scale correlation in open water and ice for Canadian Coast Guard "R-class" icebreakers*, J. Ship Res. 4, pp.249-261

SUN, Q.; ZHANG, M.; ZHOU, K.; GARME, K.; BURMAN, M. (2022), *A machine learning-based method for prediction of ship performance in ice: Part I. ice resistance*, Marine Structures 83

SUN, Q.; CHEN, J.; ZHOU, L.; DING, S.; HAN, S. (2024), *A study on ice resistance prediction based on deep learning data generation method*, Ocean Eng. 301

THOMAS, G.; KUHL, M., (2019), *Computational approaches to modeling ship interaction with ice*, Cold Regions Science and Technology, 167, pp.102-110

TUHKURI, J.; POLOJÄRVI, A. (2018), *A review of discrete element simulation of ice–structure interaction,* Phil. Trans. R. Soc. A376

VEERAMANI, M.; PERERA, P.; LI, T., (2020), *Advances in numerical methods for ship-ice interactions*, J. Marine Science and Eng. 8/2, pp.101

VROEGRIJK, E. (2015), *Validation of CFD+ DEM against Measured Data*, Int. Conf. Offshore Mechanics and Arctic Eng., New York

WANG, C.; HU, X.; TIAN, T.; GUO, C.; WANG, C. (2020), *Numerical simulation of ice loads on a ship in broken ice fields using an elastic ice model,* Int. J. Naval Arch. and Ocean Eng. 12, pp. 414-427

WANG, J.; DERRADJI-AOUAT, A. (2011), *Numerical assessment for stationary structure (Kulluk) in moving broken ice*, 21st Conf. Port and Ocean Eng. under Arctic Conditions (POAC), Montréal

XIE, C.; ZHOU, L.; LU, M.; DING, S.; ZHOU, X. (2023), *Numerical Simulation Study on Ship–Ship Interference in Formation Navigation in Full-Scale Brash Ice Channels*, J. Mar. Sci. Eng. 11

YAMAGUCHI, H.; TOYODA, M.; NAKAYAMA, H.; RHEEM, C.K.; MATSUZAWA, T.; KATO, H.; KATO, K.; ADACHI, M. (1997), *Influence of floe shape on behaviour of ice floes around a structure*, 16th Int. Conf. Offshore Mechanics and Arctic Eng. (OMAE), Yokohama

YANG, B.; ZHANG, G.; RAO, H.; WANG, S.; YANG, B.; SUN, Z. (2024), *Numerical simulation of the manoeuvring performance of ships in broken ice area*, Ocean Eng. 294

XU, P.; CHEN, B.; GUO, Y.; WANG, H. (2024), *Numerical Simulation Study on Ice–Water–Ship Interaction Based on FEM-SPH Adaptive Coupling Algorithm*, Water 16

ZHAN, D.; AGAR, D.; HE, M.; SPENCER, D.; MOLYNEUX, D. (2010), *Numerical simulation of ship manoeuvring in pack ice,* 30th Int. Conf. Offshore Mechanics and Arctic Eng. (OMAE), Shanghai

ZHAN, D.; MOLYNEUX, D. (2012), *3-dimensional numerical simulation of ship motion in pack ice,* 31st Int. Conf. Offshore Mechanics and Arctic Eng. (OMAE), Rio de Janeiro

ZHANG, B.; ZHAO, J.; WU, C. (2024) *Numerical simulation study on the ice load and ship motion response in formation navigation in brash ice,* 43rd Int. Conf. Ocean, Offshore and Arctic Eng.,

Singapore

ZHANG, J.; ZHANG, Y.; SHANG, Y.; JIN, Q.; ZHANG, L. (2021), *CFD-DEM based full-scale ship-ice interaction research under FSICR ice condition in restricted brash ice channel*, Cold Regions Science and Technology 194

ZHONG, K.; NI, B.; LI, Z.; XU, X.; WEI, H.; LI, H.; XUE, Y. (2023), *Direct measurements and CFD simulations on ice-induced hull pressure of a ship in floe ice fields*, Ocean Eng. 272

ZHONG, K.; LI, Z.; NI, B.; XUE, Y. (2024), *Numerical Studies on ice-induced loads on ship hulls in ice floe regions,* 43rd Int. Conf. Ocean, Offshore and Arctic Eng., Singapore

ZHONG, N.; POTENZA, A.; SMITH, S. (2024), *Autonomous Navigation in Ice-Covered Waters with Learned Predictions on Ship-Ice Interactions*, arXiv preprint arXiv:2409.11326

ZHOU, L.; SUN, Q.; HAN, S.; WANG, A. (2023), *A Machine-Learning-Based Method for Ship Propulsion Power Prediction in Ice*, J. Marine Science and Eng. 11

ZOU, M.; ZOU, Z.; ZOU, L.; CHEN, C.; ZHANG, X. (2024), *Numerical investigations of the ice force distributions on a ship in brash ice channel using a simplified CFD-DEM solving framework*, Ocean Eng. 312

Digital Twin combined with XR in Design, Training & Operation

Helge Bjordal, Nagelld, Bergen/Norway, helge.bjordal@nagelld.no

Abstract

This paper explains through customer stories how Sensor data and Extended Reality (VR & AR) is applied effectively as collaborative tool for design review, operation and realistic training. As an illustrative example, in a boat design, more than 30 design flaws were found when using VR in the design project. Another example application uses VR in a man-over-board operation training. Sensor data paired with Digital twins can be used to train using live data and Operations department can experience how the vessel is operating 24/7. Digital twins can be used to optimise operation workflow and to investigate accidents.

1. Introduction

Digital transformation technologies are major topics in the maritime industry today. But to utilize these new technologies, we need to design the vessels "correctly", train crews to operate the technologies correctly and plan optimal maintenance schemes. How can this be done in a way that doesn't pollute our environment, saves time and money, while ensuring correct HSE with zero damage to personnel and equipment and opens opportunities for realistic collaborating worldwide without time-consuming travels and emissions to the environment?

Fig.1: A "member" in our VR test panel – our XR lead's daughter testing a VR application

2. Virtual Reality (VR) and Digital twin explained

To fully understand this paper, it is necessary to explain what this paper means by VR. VR is short for Virtual Reality, which means just that. When you put on the VR equipment you are "transported" from your normal reality and into a virtual reality. In this virtual reality you can communicate with other virtual participants and execute tasks. The other virtual participants may be human, like yourself, or digital avatars that are pre-programmed or are AI controlled. The tasks can be designing vessels, training onboard, planning vessel operations and so on.

Most important, VR itself has no limits to what you can do and how you can operate. Limits are created by the technology, budgets, timelines and you imagination. More on that later in this paper under the headline “Why VR is the best”.

VR is a part of the XR (Extended Reality) family, which is VR, AR (augmented reality) and MR (mixed reality). VR is already explained in the beginning of this chapter. Meta Quest 3, Pico and Varjo is such products.

Fig.2: Meta Quest 3 VR set

AR means a combination of reality and graphics, where the graphics doesn’t interact with reality. Example: If you “throw” a 3D object into the room it just go “through” any real object, like a table, a chair or even a person without interacting with the objects. XREAL One is such a product.

MR means a combination of reality and graphics, where the graphics interact with reality. Example: If you “throw” a 3D object into the room it will interact when it hits a real object, like a table, a chair or even a person. Microsoft HoloLens and Meta Quest 3 is such products.

A typical VR set consists of a head mask, often called “goggles” and two hand controllers. The VR set can communicate with other VR sets via Wi-Fi or cabled to a computer. The computer then communicates with other VR sets directly, via Wi-Fi, or via another computer with a VR set attached to it. The VR set attached to a computer can use cable or wi-fi, i.e. you always can work “cable-free”.

Fig.3: Modern VR sets are easy to use – no cables required!

Different VR sets have different features and different technical specifications. The most obvious one is visual quality. The higher the resolution of the lenses in the VR set, the more realistic the VR experience can be. However, price and resolution are not necessarily comparable. VR sets also have different amount of storage and different CPU's which influences on the price. VR sets have a price-range from €350 to more than €10 000! Usually, you content deliverer will give you good advice regarding which VR set to choose.

The VR sets can contain the software it runs itself, via the computer it is attached to, via a cloud-based service or via a server located somewhere on the internet. Let's investigate the pros and cons by showing some types of projects that are up and running today.

Digital twin, in its most used form, means that you have a digital (3D) model of a real asset. Typically, this can be a 3D model of a sailing ship, like the Odfjell vessel shown in figure 4. More complex digital twins are 3D models connected to a data-stream from one or more sensors onboard the actual sailing ship. This can be used to plan operations, when to perform maintenance or investigate accidents. More on this later on in this paper. Let's look at digital twins without the sensor feed first.

3. Design reviews using VR when planning to build a vessel

Digital ship drawings are huge 3D models. This requires a computer to "crunch" the data before it can be displayed in VR. Typical setup is that all the different participants that are connected to the design of the vessel runs a software, like Spawn, where the data is stored in the "cloud" and shared between the design team (and maybe the vessel owner and future crew members).

4. Planning operations using VR

When a vessel has completed the design phase you may use the vessel drawings to start planning how to operate the vessel.

We still have huge drawings, but these can be optimised to run directly in the VR sets if you want to avoid the "hassle" using a PC running VR software, like Spawn.

Fig.4: Odfjell vessel "Bow Hercules» in VR

4.1. Onboarding new crew members using VR

Your new vessel is to be manned (maybe not in the future). Your existing vessel drawings in VR means that your crew can familiarize itself with the ship, knowing where everything is placed, muster stations, types of equipment used etc. You can still choose to use the "original" huge 3D drawings, run via computers, to the VR sets, but I recommend optimising the drawing so that everything can be stored inside the VR set.

If you want your crew to cooperate during this familiarization phase you can have a server connecting all the different VR sets and distribute data to the different participants. The crew members can be located on different physical places but still work together.

4.2. VR based training onboard a sailing ship

Some months ago, this was impossible because we found that the VR set developers never intended their VR sets to be used in an environment that was moving (sailing). Nagelld was the first ever to encounter this problem when we were making a lifeboat VR training. This is now fixed. Meta calls this feature "travelling mode" and it must be checked off inside the VR set.

The lifeboat crew now has a local "internet" where a server and several VR sets communicate so that the crew can run training sessions when the vessel is sailing.

We are now testing how it will be to combine VR sets onboard vessels running the Starlink satellite broadband with VR sets ashore. If this works out well, you will be able to run training sessions with sailing crews connected to land-based experts.

4.3. Implementing VR in a customer organisation - Logistics of hardware and software

First an important advice: Your VR project is not an IT department project! You may need the IT department to buy and implement the VR sets and computers into your company network, but the IT department cannot help you choose VR sets, servers or computers needed! To choose correct equipment specifications, you need to talk to your VR content supplier. I cannot stress this enough!

Depending on how many VR applications and VR sets you need, you also need a plan to support and deploy the applications and sets. Most VR suppliers have software to handle this logistics, but many of the solutions are bad and they have a tendency of changing depending on who owns the company. An example here is how the Oculus for Business software was discontinued when Meta bought the company. Meta also made the VR set impossible to use in China because of the Facebook connection.

Several independent logistic solutions are available, like ManageXR and ArborXR. Talk to your content provider or connect to a VR cluster, like VRINN in Norway, they will advise you independently.

5. Research & Development

The most recent R&D projects in VR are "hand tracking", "see-through VR" and a combination of both:

- "Hand tracking" means that when you are learning how to operate any type of equipment you don't use the VR controllers anymore, you use your hands instead. Example: To push a button, you simply push the virtual button using your finger and not by pressing a selection button on the controller first. This means that you can "pick-up" tools, "turn" levers, "push" buttons and so on just like you would have done in "real life". This makes the VR experience to become much easier to use.

- "See-through" VR means that you can switch between a complete VR experience and a MR experience where 3D objects interact with the real surroundings. Example: you want to "isolate" a valve so you can concentrate on it without having to interact with the rest of the engine room. Maybe you want to call in a land-based expert to show you how to fix the valve and it is easier when the particular piece of equipment is isolated and the expert actually can see both the "real" valve and the 3D model of the valve.

Combine the "see-through" with the "hand-tracking" and you can fix the valve using correct tools directly with your hands in an isolated view. New technology will open for new content which again will open for new ways to work and interact in VR.

6. Why VR is best

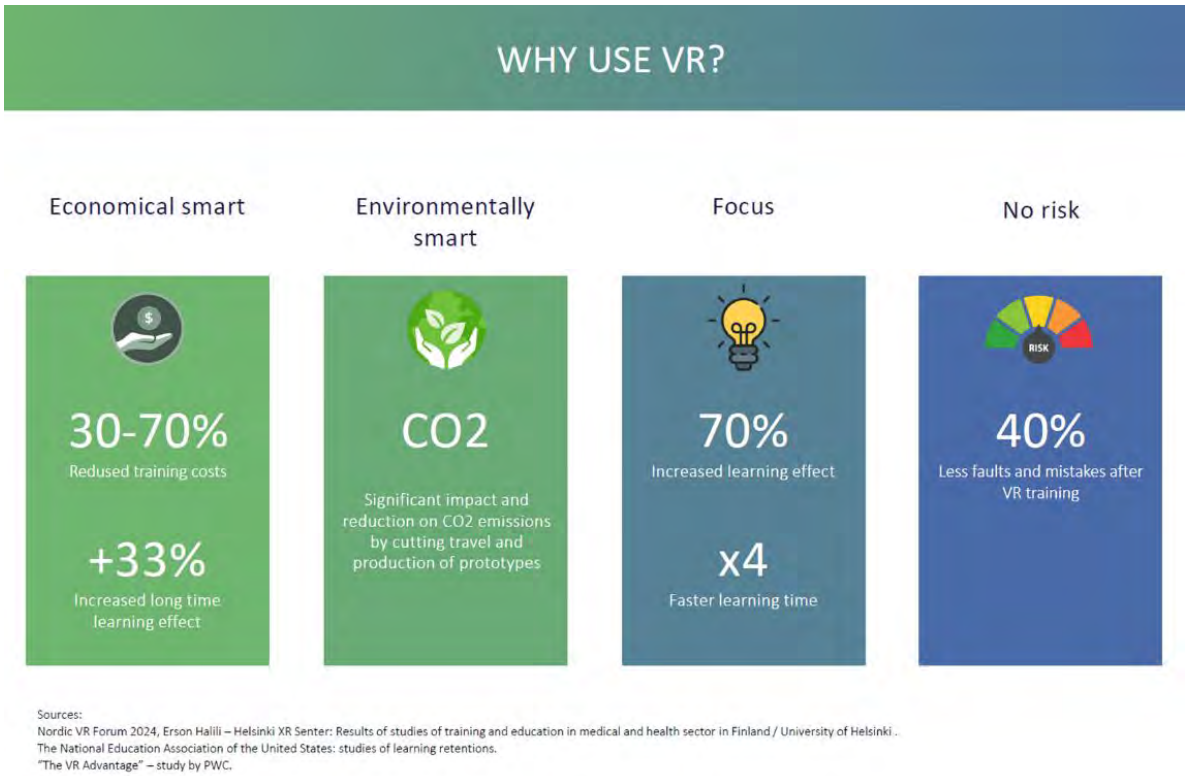


Fig.5: Why VR is best – data from different studies

Would you like a work environment without any physical limitations in space and time, where you can assemble your team instantly and independent of current physical location of the team members, where nobody is injured and your equipment never is damaged even when your team is training on accidents that may happen. A place where everybody is 100% focused only on the tasks at hand and you can evaluate everything that has happened. Where you can reset your world and begin again, over and over. Where you can test multiple solutions and design without scrapping any metal. Where you and your sub-suppliers can meet and discuss, test and experience different solutions until they are 100% optimal? Where your crew can train on operations and maintenance over and over.

This world does exist today, and you don't even have to swallow a pill to get there! Let's take a closer look at the different parts that makes up this work environment.

6.1. No physical limitations

I call this feature the "Matrix syndrome". In virtual reality you are only limited by the feature of the VR presentation. You can jump 50 m at the time, you can fly, you can lift any structure, you can poke

inside closed gear boxes, you can examine a vessel's running engine – from inside the engine! But you can also be restrained to the world as we know it. Your choice!

A true story related to the subject: A company was ordering a new RIB (rigid inflatable boat) and wanted the crew members to "experience" and comment on the design before it was actually built. We set up a VR experience where the crew members could get a "look & feel" of the boat. The crew found a lot of stuff they wanted to change which resulted in a completely different design based on the operations the crew wanted the boat to carry out.

During the VR experience one of the crew members asks me how he can "physically" look into the hull structure of the boat, maybe we can remove the outer hull or make it transparent, he suggests. The other crew members had similar suggestions. No one suggested to simply put one's head through the outer hull!

Fig.6: RIB vessel in VR

This experience made me realise that most people are like Neo in the Matrix. We are always limited to what we think is possible and not what is actually possible. This is something we must train to overcome when creating VR presentations. (By the way: are you quite sure you can't put your head through the nearest wall?)

6.2. Multi-user collaboration independent of physical location

Covid – yes, I need to mention it. It separated us from working and meeting physically. In VR we can meet and work together without any chance of contagion because we don't have to be physically located in the same place. We have tested VR several countries and even continents apart. One company has VR rooms in Singapore, Houston and Bergen where people meet to collaborate. This also gives a friendly nod to our environment. You no longer have to make those 2-days-travel-1-hour-meeting journeys. Just meet up in VR.

6.3. No injuries to people and no damage, wear and tear or accidents on your machines

Let's look at cost and ROI (Return On Investment). It costs to develop VR training programs, but the more interesting thing is to look at the ROI. How fast is the investment paying off? Not just in actual costs related to a training session, but also the cost of injuries and machine wear and tear. You can even train on scenarios that are impossible to train on "live", Like capsizing, explosions and severe fire.

One company has a cost of € 50000 on every training session. They may have 1-2 sessions each week. The entire VR training simulator costed € 150000. The training sessions are offshore and involves expensive equipment. Each crew trains maybe 10 times per year. For now, they don't train less in physical training, but they train correctly every time because the crews can train in VR as often as possible, honing their skills. The different crews, located along the entire Norwegian North Sea/ Norwegian Sea basin also have the possibility to train together, learning from each other, without needing to meet physically. Now each crew member can train 5-10 times a week.

6.4. 100% focus

Training using VR ensures 100% focus on the task at hand. No disruptions from your mobile phone or other people located around you. No mail to be answered or web pages to be scrolled. VR is so focused that we had to implement a clock showing the real-life time in our VR software. I have been late to countless Teams-meetings because VR erase your internal "clock".

Every time our customers have experienced VR together with us, I ask them about how long time they think they were in the VR experience. So far, no one have been close. I'll share a story for illustration: Some engineers from a customer of ours came into one of our offices to test out a new pump design using our VR equipment. One of the engineers was not too keen on using VR because she had a bad experience earlier (Euro Disney VR experience!). I talked her into trying and within few minutes she was so focused on the task that she completely forgot that she was using VR. When they had tested what they wanted (and found that they had to re-think the entire maintenance operation) I asked her how long she thought she had been in the VR experience. 15-20 minutes maybe, she answered. The correct time was just over one hour!

6.5. True evaluation

VR doesn't allow cheating and shortcuts, if we don't want it to. We all know that walking/standing under a hanging load is a big no-no in the real world. In VR you don't get away with doing that or any other shortcut. The VR experience monitors everybody always and can give you a complete evaluation of every training.

7. VR in design

Do you know the series called 'Deadliest catch' where they fish for snow crab? A Norwegian company is developing such a vessel with a 'clean deck' operation. The vessel has two main cranes. The crane supplier wanted a VR project where the customer could run the cranes in VR to make sure the cranes operated within the specs given. We made the VR project and all seemed fine.

The vessel supplier wanted us to include the entire operation, not just the crane handling. So we set up hatches and winches - and hey! One of the cranes interfered with a hatch in one given point of the operation rendering the entire operation void! We were asked to re-model the hatch, so it didn't interfere. After a new 3D model of the hatch was modelled and implemented by us, everything was fine. I asked them how much they saved and they wouldn't say! This, and many other stories makes we wonder how much time and money is mis-spent when designing a vessel.

There are many considerations to make when designing a vessel. Why not use VR to make all these considerations play together or at least know where the issues are? One of the considerations to make in today's vessel design is which "fuel" that are to be used. The different "fuels" require completely different vessel designs. Whether the vessel is hybrid fuelled, electrical, sails, ammonia, hydrogen, or LNG, to mention some alternatives. Designing in VR makes it possible to "test" different fuel lay-outs in your vessel design.

In Norway we had an accident where the pilot didn't see the vessel he crashed into because the beam structure hindered view from the pilot seat in some angles. This would never have happened if the

pilot could “test” his viewing angles in the design phase using VR because VR gives 1:1 view of your surroundings.

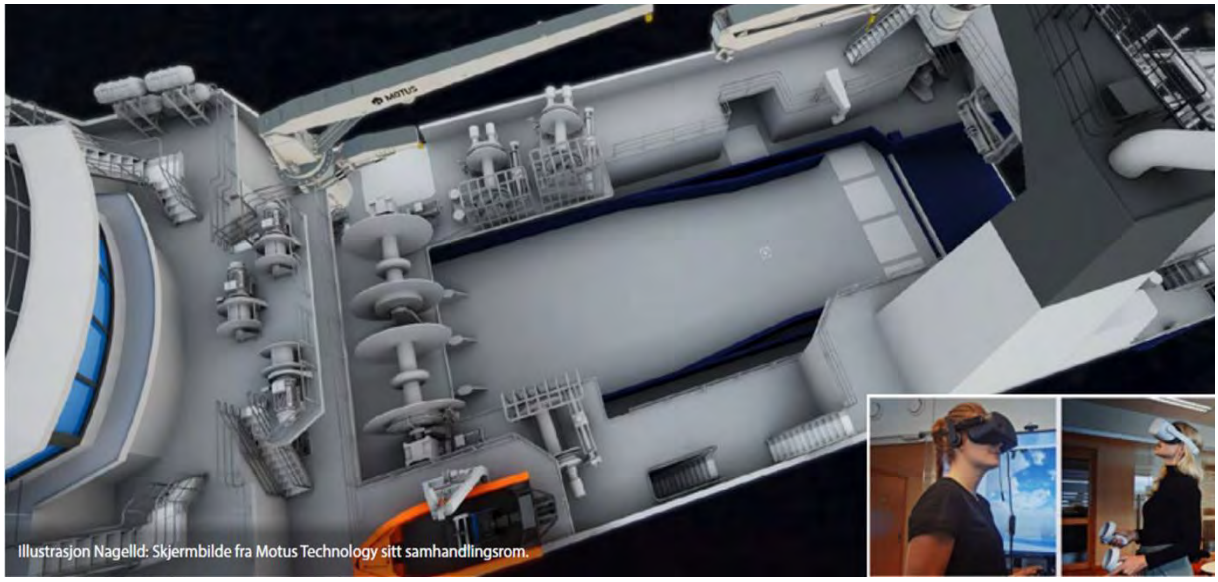


Fig.7: Snow crab vessel in VR

All kinds of layouts can be tested and optimised using VR in the design phase:

- How is the accommodation for the crew
- How many containers can my vessel handle and are the container locks placed correctly
- Is the view from the bridge optimised according to work operations
- Does the equipment from sub-contractors connect correctly
- How easy is it to replace or maintain a larger part within the vessel’s hull
- Etc. etc.

If you design a vessel today without using VR Design Reviews, chances are 99% that you are wasting time, money and the environment. 100% of all projects, not just ship designs, we have run through VR have had design faults or unwanted operational / maintenance design.

Remember also that a digital 3D model of the ship in VR opens up for other departments to start working with the vessel much earlier than before:

- Operation planning
- Maintenance planning
- Onboarding of crew members
- HSE planning
- Recruiting (HR)
- Showing the vessel to potential customers
- Showing the vessel in exhibitions
- Making a digital twin of the vessel

8. VR in training

The challenge: We want to train a MOB (Man Over Board) crew on operation and communication before we put them in a real boat? The solution: Make a MOB boat training simulator that uses real physics on the 3D models. This simulator requires a rather powerful computer connected to each of the three crew VR sets because the VR sets cannot handle the real-time physics that is calculated when the MOB boat flies across the virtual waves! The simulator trains the crew in communication,

operation (steering the boat in all kinds of weather and finding the missing persons) and maintenance. An operator “throws” trouble at the crew in real time, the operator also has the tools to evaluate the session both in real-time and afterwards.

Fig.8: MOB boat in its cradle

Training you personnel using VR is the best way of training, by far! The reasons why we claim this to be true is from messages we get back from our customers. After training in VR, they solve the tasks correctly the first time without wasting any time on pondering how to solve the task and they remember how to solve the task at hand. Using hand-tracking in the VR training can be very realistic because you must pick up the correct tools and use them correctly to solve the task at hand. Just like you would do in real-life.

Star Trek: Kobayashi Maru - When training in VR you can also train “impossible” scenarios or accidents, just to see how your crew react to difficult or no-win scenarios.

Most important when, training in a VR simulator, is that your crew can train over and over, by themselves or with other crew members.

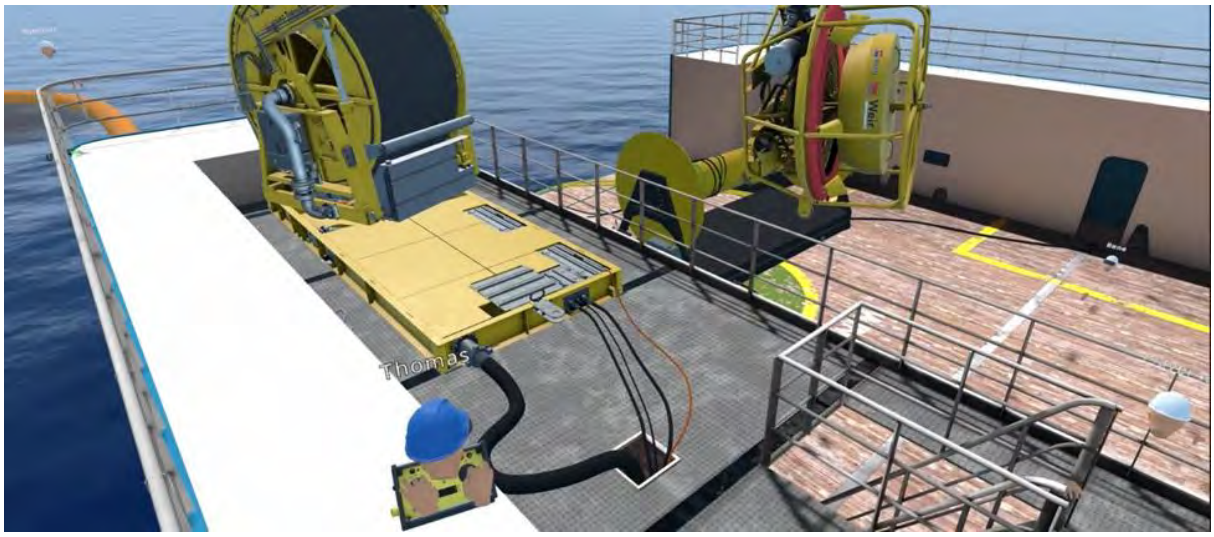


Fig.9: VR collaboration in the NOFO Offshore Oil Spill simulator

9. One-way and two-ways sensor data driven Digital Twins

This paper has described digital twins of chemical ships, MOB boats, lifeboats and offshore oil spills where the user has interacted with the 3D model. What if we connected all the sensors from the "live" ship and connected it with the equivalent digital 3D models?

We could enter the digital twin and look for wear and tear on parts. We could follow all kinds of movements onboard, from engine parts to deck cranes. We could look for ways to optimise an operation based on real data from the ship. The Starlink systems makes it possible to send "live" sensor data from ship to shore 24/7!

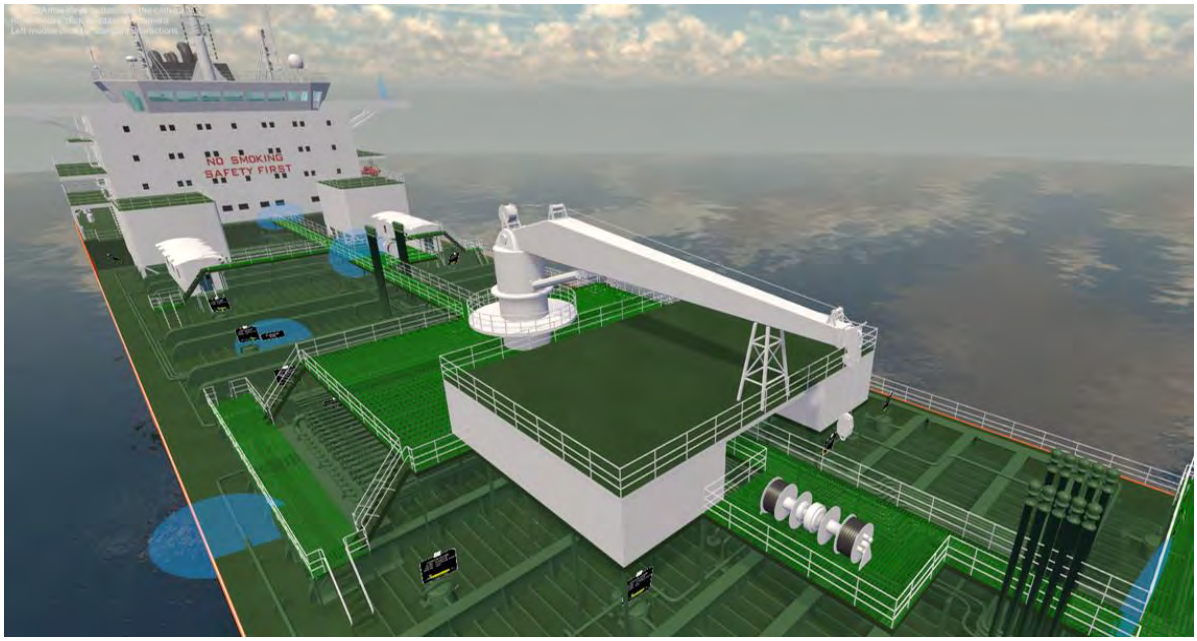


Fig.10: Odfjell digital twin showing data regarding the cargo in each tank

Data collected from sensors combined with 3D models would give invaluable input in how to design even better ships in the future. Sensor data compared with simulated data of the same ship will calibrate simulations to be even better in the future. The different crews can "compete" in delivering the best sensor data to the digital twin, showing how optimal the ship can be run. These sensor data can also be used to train and operate strategic and tactic. That means that data isn't just coming INTO the digital twin but is also sent OUT from the digital twin to the actual vessel or fleet of vessels to advice or control the vessel. This will especially be the case where the vessel is autonomous.

Fig.11: Strategic and tactical operation/training where the collaboration is via VR.

We are discussing a complete strategic/tactical virtual “control-room” with a large customer. This means that data can be sent both IN and OUT via the digital twin “control-room”. The “control-room” will be digital twin of a physical control-room. The digital control-room will be accessible via smart-phones, VR and PC’s which means that the participants can be located anywhere as long as they can access a digital platform with wi-fi connection. Future use cases can involve digital twin “control-rooms” controlling entire vessels and fleet of vessels.

10. Extra material

I have often encountered companies and persons that want to buy VR but are stopped because they lack what I call “competence of XR project purchasing”. That’s why I have added a short helper when purchasing a VR project:

- What issue is the project going to solve? - The developer needs to know the issue(s) you are trying to solve in order of manage to price the project. If the project order is “fuzzy” it can become expensive really fast.
- How much does the issue cost me today? - Is the issue solved today? What is the cost today? These are important questions when forming a budget.
- What ROI can I expect? - The developer should be able to explain the possible ROI of the project. Also beware that some VR projects may open up new business models for your company.
- How is the developed project priced (licensing or one-off payment with support)? - Do you want to pay a license fee per person, not owning the solution or do you want to pay “one-off” and own the solution. You usually pay a yearly support fee if you buy a one-off.
- Off-the-shelf solution or a developed solution? - Off-the-shelf solutions usually gives you 80% of your solution, if you can find such a product. A developed solutions cost much more but give you control of the content of the product.
- Who owns the IP of the finished product? - This depends on the payment model and how exclusive you want to own the product.
- What equipment is needed (VR sets, PCs, etc.) - The developer can spec the equipment/hardware you need. Buy the equipment from your local dealer. In that way you get help fast if something stops working.
- Single user or multi-user (or both)? - Do you want to collaborate in VR or not. Multi-user is more expensive than single user and usually requires more hardware. The upside is that your team can collaborate directly in VR.
- Who are supplying the digital 3d models needed (customer, sub-contractors, modelled by the VR project supplier)? - No 3D models mean no VR project. Who is going to supply the 3D models needed? If you work with a developer that has a lot of 3D models it is usually less expensive than to work with a developer that need to model the 3D models not provided by you or your sub-contractors.
- What is the expected timeline? - Always agree on a timeline with milestones. This keeps everybody in the project on its toes and ensures forward momentum. Remember that it is usually the customer and not the developer that slows the project down…
- How much involvement is needed/expected from the buyer? - The timeline with milestones will decide this. Working with a developer that knows the industry usually requires less involvement from the buyer.
- Who is testing the product? - Testing takes time and money. Decide on a test plan as soon as possible.
- Who is in the project team and why should you avoid your IT department? - Put together a team that controls the financing of the project, a product expert / tech. person and the person with the problem. IT departments are only to be involved if needed regarding network security and purchasing of PCs. Usually, the developer and the IT department wants different hardware specs because the IT department wants to streamline all computer equipment, and the developer wants you to buy the equipment that serves the VR project best.

Acknowledgements

All VR work presented in this paper is developed by Nagelld. Nagelld is recognized as one of the frontrunners of VR experiences to the maritime industries worldwide. One of our VR training scenarios won “The Jury Award” at Nordic VR Forum 2024. In 2023 we won “Best VR training experience” at Nordic VR Forum for a cruise ship lifeboat VR training experience. Nagelld has also developed Spawn, a metaverse collaboration platform for design reviews and various operation and training scenarios.

The data presented in figure 5 is extracts from Erson Halili / Helsinki XR Center – *Results of studied of training and education in medical and health sector in Finland*, The National Education Assocoation of the United States – *Studies of learning retentions* and PWC – *“The VR advantage”.*

Vessel-Based AIS

Enes Bicer, Wärtsilä, Helsinki/Finland, enes.bicer@wartsila.com
Dennis Grebasch, Wärtsilä, Helsinki/Finland, dennis.grebasch@wartsila.com
Kay Dausendschoen, Wärtsilä, Hamburg/Germany, kay.dausendschoen@wartsila.com

Abstract

This paper presents a simulation-based evaluation of vessel-based AIS stations as an alternative to terrestrial AIS, limited to coastal range, and satellite AIS, which suffers from signal collisions and degraded performance in dense traffic regions, for maritime traffic monitoring. The AIS data market is also consolidating, raising concerns over accessibility and cost. We evaluate the number of vessels equipped with AIS receivers required to achieve near-continuous global coverage. Results show that while 2,000–3,000 station vessels are sufficient to detect over 75% of IMO-registered traffic once per day, achieving broader hourly coverage requires larger fleets beyond 6,000 vessels.

1. Introduction

Considering that 90% of global trade by volume is conducted via sea, monitoring ships is crucial. It supports vessel navigation and helps avoid critical situations such as collisions, oil pollution and other maritime accidents. Effective traffic management is also essential, especially in congested areas, *Fournier et al. (2018)*. With a yearly increase of 2.4%, maritime trade volume which is typically measured in tons or ton-miles reached approximately 12 million tons in 2023. In parallel, total ton-miles grew by 4.2% to around 62 billion. The global fleet of ships over 100 gross tons included roughly 110,000 vessels, encompassing both cargo and non-cargo ships, *UNCTAD (2024).*

Given this scale and growth, reliable monitoring technologies are indispensable to ensure both safety and efficiency in maritime operations. One of the most widely adopted systems serving this role is the Automatic Identification System (AIS). Although radar helps detecting vessels, it doesn't provide information about identity and intention, but AIS allows vessels to both identify and be identified, *Robards et al. (2016)*. It was developed in the 1990s with the aim of preventing ship collisions and enhancing navigational safety. If a vessel equips with AIS, it can transmit and receive information. It enables communication with other vessels and long-distance communication with coastal authorities, *Yang et al. (2019)*. AIS has become mandatory for all vessels more than 500 gross tonnage, vessels both more than 300 gross tonnage and engaged in international voyages and all passenger vessels regardless of size with the revision of Chapter V, Regulation 19 of the Safety of Life at Sea (SOLAS) Convention in 2000. Vessels meeting these criteria had to equip with AIS latest by 31 December 2004, *IALA (2004).*

While AIS has been extensively studied in terrestrial and satellite-based applications, shipborne devices referred to here as vessel-based AIS stations remain less explored. Terrestrial stations are limited to coastal areas due to range constraints, and satellite-based AIS still faces challenges such as message collisions. In contrast, vessel-based AIS offers the potential for dynamic and continuous coverage. This potential has become even more relevant in recent years as satellite communication costs have declined, making it increasingly feasible for vessels to transmit data continuously. Understanding the scalability and impact of such systems is essential to addressing existing monitoring gaps and improving maritime data collection.

Building on this research gap, the present study evaluates the continuous coverage capabilities of vessel-based AIS stations. Specifically, it investigates the following questions: (1) Can vessel-based AIS stations provide adequate coverage of global vessel traffic? (2) How many vessels equipped with AIS receivers are necessary to achieve comprehensive and efficient coverage? (3) Does coverage effectiveness decrease as fleet size increases, and at what point does adding more vessels result in diminishing returns?

This research contributes to the maritime domain in the following ways: (1) It presents a simulation model for evaluating vessel-based AIS coverage which later can be used and adapted to broader maritime research applications. (2) It provides insight about vessel-based AIS potential in maritime traffic monitoring.

2. Literature Review

AIS is a communication technology developed to prevent collisions, increase maritime safety and help vessel traffic management. By using Very High Frequency (VHF) radio channels, it facilitates vessels to exchange data consisting of real-time information about vessel identity, position, speed, course, etc., *Dong et al. (2019)*. AIS-equipped vessels can exchange data within the range of VHF autonomously and continuously which improves situational awareness, traffic management and maritime safety. It also supports various safety and environmental objectives such as search and rescue operations, vessel activity monitoring for security purposes, etc., *IMO (2015)*.

Over time, AIS has become a cornerstone of the digitization of the shipping industry. Archived AIS data have been helping to many research applications which goes beyond navigation from emission monitoring, trade flow estimation to marine conservation, *Dong et al. (2019)*. This evolution has established AIS as an indispensable technology for improving maritime efficiency, safety and environmental protection, *CMTS (2019)*.

Given this wide range of applications, effective data collection is essential. AIS data are gathered through a distributed network of receivers, known as AIS stations, that capture and relay vessel information for further use. These stations, identified by unique Maritime Mobile Service Identity (MMSI) numbers and operating under an international standard, *IALA (2016)*, include both shipborne devices and fixed, shore-based (terrestrial) units. Together, they form the backbone of AIS infrastructure and determine the reliability and completeness of global maritime monitoring.

2.1. Terrestrial AIS

T-AIS (Terrestrial AIS) is one of the types of AIS stations. It refers to the network of land-based AIS receiving stations and receive the AIS-transmitted information from vessels withing their range periodically. The network reach across thousands of ports and coastal shipping routes globally as can be seen in Fig.1. The typical range of a terrestrial AIS station is 15-20 nm for an external antenna placed about 15 m above sea level. However, this range can reach up to 40-50 nm depending on the altitude and type of the antenna with the affection factors such as nearby obstacles and weather conditions, *MarineTraffic (2021)*.

T-AIS is the optimal solution to monitor maritime traffic near ports and busy shipping lanes because it is effective in coastal areas in terms of update frequency and detection rates. However, since it has a lower coverage range as highlighted before, it cannot reach vessels in regions beyond coasts such as oceans and remote waterways, *MarineTraffic (2021)*.

2.2. Satellite AIS

Due to the curvature of the Earth, ship-to-ship communication is limited to ~20 nm, depending on antenna height, *Hasbi and Kamirul (2020)*. Similarly, ship-to-shore communication is constrained to a range of 40 nm as highlighted in the previous section. To extend ship detection coverage, satellite-based approaches were developed around 2004 by military forces, maritime spatial research organizations and the industry, *Fournier et al. (2018)*.

The satellite-based Automatic Identification System (S-AIS) utilizes Low Earth Orbit (LEO) satellites equipped with AIS transponders for ship surveillance. Since VHF radio waves can reach altitudes of up to 1000 km, LEO satellites can receive VHF signals transmitted from AIS-equipped ships. These transponders collect AIS data and relay it to ground stations. The AIS satellite system comprises sev-

eral components, including LEO satellites, shipborne AIS equipment, users, communication links and ground stations, *Chen (2014).*

LEO satellites travel at speeds of approximately 27,400 km/h and maintain altitudes ranging from 650 to 800 km above the Earth's surface. Each satellite completes a full orbit in roughly 90 minutes and that means there is a need for constellation of satellites to ensure continuous coverage, *IALA (2011).* The S-AIS system can monitor vast areas, with a potential coverage of up to 15 million km², *Hasbi and Kamirul (2020).*

T-AIS systems provide near 100% detection rates for ships near the shore and offer continuous coverage. However, they are limited by range and high infrastructure costs. In contrast, S-AIS offers global coverage that bridges the coverage gaps in remote coastal areas and provides more cost-effective way of tracking per square mile. Nonetheless, S-AIS has lower detection rates near the shore compared to T-AIS. While T-AIS is primarily used for collision avoidance, S-AIS is valuable for tracking vessel behavior. Thus, both T-AIS and S-AIS are essential for comprehensive maritime domain awareness and it is critical to integrate T-AIS data with S-AIS data, *IALA (2011).* As also can be seen in Figure 1, in terms of coverage range, it is essential to integrate T-AIS with S-AIS.

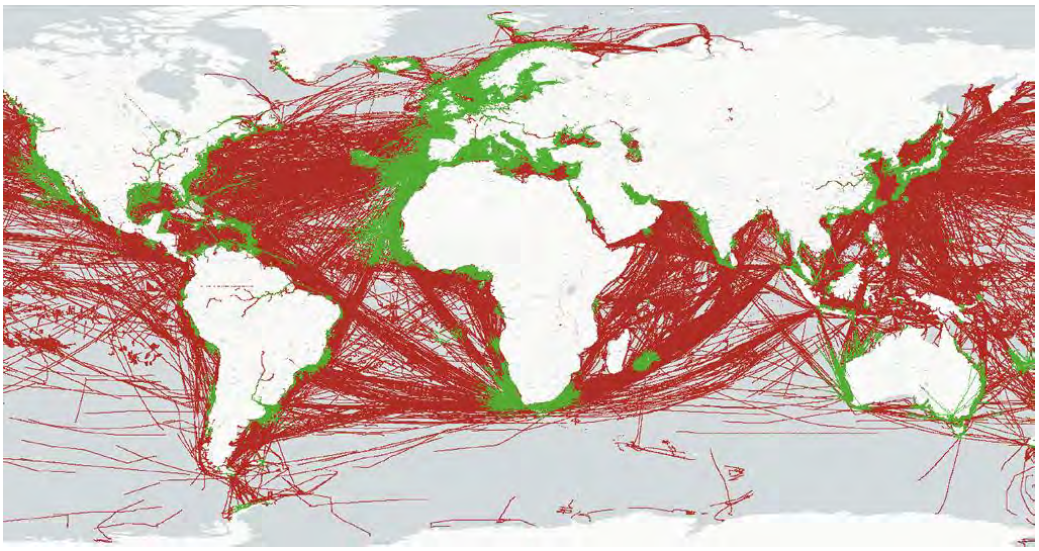


Fig.1: MarineTraffic terrestrial (green) and satellite-based (red) AIS network coverage, *MarineTraffic, (2021)*

2.2.1. Key Limitations and Challenges

Although satellites provide high detection rates in areas beyond the reach of terrestrial receivers there are several challenges exist. These include difficulties in receiving lower-powered AIS-B transponder signals, satellite revisit times and data quality issues, *Fournier et al. (2018)*. Additionally, *Cervera et al. (2010)* identified technical challenges such as message collisions from ships transmitting from different SOTDMA cells, terrestrial interference from fixed and mobile systems, Doppler effects, propagation channel delays, lower signal-to-noise ratio, etc.

2.1.1.1. Message Collision

SOTDMA is primarily used by mobile stations operating in autonomous or assigned modes. To prevent conflicts and ensure continuous and predictable messages transmission, this protocol employs a slot reservation. It dynamically selects time slots based on information from surrounding stations. In this way interferences are minimized and systematic position and identification data transmission are enabled, *ITU-R (2014).* A satellite at an altitude of 650 km covers an area with a diameter of ~5000 km, allowing multiple SOTDMA regions to be monitored by a single satellite. However, as ships independently organize their time slots, stations in different SOTDMA regions that are out of each

other's range cannot coordinate slots, leading to message collisions as seen in Fig.2, particularly in high-traffic areas *Cervera et al. (2010).*

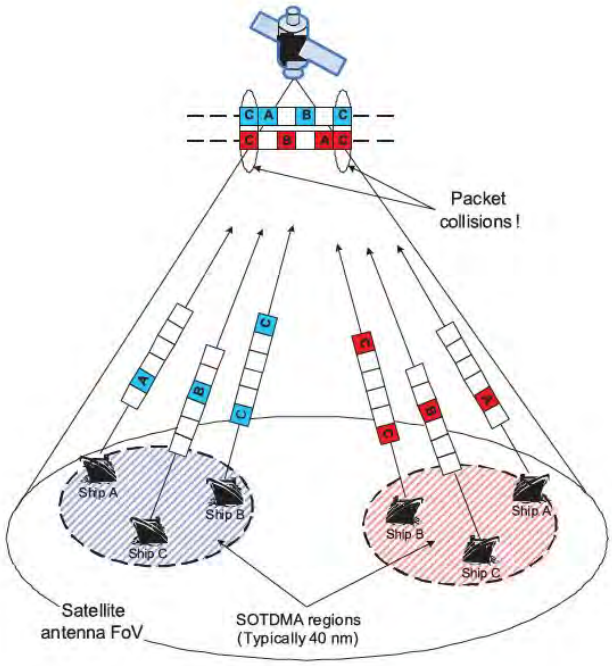


Fig.2: Satellite packet collisions, *Cervera et al. (2010)*

In the satellite view, when the number of ships exceeds 1000, signal collisions increase, reducing the number of successfully received messages. When the number exceeds 2500, collisions become critical and very few uncollided messages remain. The AIS channel frame has a capacity of 4500 messages per minute. In a region with 1000 ships, a 15-minute observation ensures that at least one uncollided message is received from each vessel due to signal scheduling. However, global ship traffic analyses indicate that areas that a satellite covers with fewer than 1000 AIS transmitters are rare, *IALA (2011)*.

To address slot collision issues, two methodologies On-Board Processing (OBP) and Spectrum Decollision Processing (SDP) are currently in use. OBP employs highly sensitive receivers similar to terrestrial AIS receivers, reducing latency between satellite pass, ground station data download and internet backhaul to the end user. However, OBP has a lower detection probability when the number of ships in a satellite footprint (approximately 5000 km in diameter) approaches 1000, *IALA (2011)*.

SDP, on the other hand, uses receivers that detect and digitize the RF spectrum, process raw spectrum files to control the noise floor and reconstruct collided messages using specialized software algorithms. While SDP requires ground-based processing which increases latency, it offers superior first-pass detection performance that enables a comprehensive maritime picture within two satellite passes, *IALA (2011).*

Channel Propagation and Interference

Co-channel interference in high-density shipping regions such as the Mediterranean, China Sea, North Sea and Gulf of Mexico reduces the performance of satellite AIS systems. Additionally, land-based interference from coastal areas can further degrade S-AIS performance by raising noise levels and lowering signal-to-noise ratios, making AIS reception unreliable, *Skauen (2015),* citing *Eriksen et al. (2006), Re et al. (2012).*

Long-range AIS was developed to mitigate interference with traditional AIS systems while extending

coverage. It operates on Channels 75 and 76, separate from regular AIS channels (AIS 1 and AIS 2), ensuring minimal disruption to standard AIS communication. Long-range AIS transmissions are also designed to be intermittent, with updates occurring infrequently, reducing the likelihood of interference with continuous AIS data exchanges, *ITU-R (2014)*. However, although 75 and 76 channels have improved tracking capabilities, land-based interference remains a persistent challenge, *Skauen (2015).*

2.3. Vessel-Based AIS

Simultaneously, the AIS unit listens on the same channels for incoming messages from other ships. This allows a vessel to collect AIS data from all surrounding traffic within range. In essence, each ship serves as both a transmitter and receiver: broadcasting its own data while plotting nearby vessels on an electronic chart or radar-like display for collision avoidance and traffic monitoring, *MarineTraffic (n.d.)*.

While AIS was designed for direct over-the-air exchange, modern connectivity allows ships to forward AIS data via the internet. Many vessels today have onboard computers or satellite communication links that can take the AIS data stream (which the ship is already receiving from nearby traffic) and transmit it to central databases on shore. Near the coast, a GSM internet connection is suitable, while a satellite internet connection is required for global coverage in the open sea. In effect, a ship can act like a floating AIS receiver station – collecting signals from its vicinity and then sending that data to an internet server whenever a connection is available, *MarineTraffic (n.d.)*.

This practice underpins several crowd-sourced AIS networks run by maritime organizations and enthusiasts. For example, services like MarineTraffic, AIS Hub and others aggregate data from thousands of volunteer AIS feeds: not only coastal antennas, but in some cases AIS receivers on ships or yachts that upload what they receive.

By combining these contributions, such platforms achieve broader coverage than fixed stations alone. A ship at sea equipped with an AIS receiver and a satellite internet uplink could, in theory, provide live AIS data for any other vessels in its radio range, extending the visibility of maritime traffic into mid-ocean areas.

Table I: Summary of the key advantages and disadvantages of different AIS sources

AIS Type	Advantages	Disadvantages
Terrestrial AIS	- High detection rates near coastlines and ports - High update frequency - Low latency - Excellent for collision avoidance and port monitoring	- Limited range (15–50 nm) - Cannot cover open ocean - Requires costly land-based infrastructure
Satellite AIS	- Global coverage, including remote areas - Cost-effective per km² for wide regions - Valuable for monitoring ship behavior and remote activities	- Lower detection near coastlines due to terrestrial interference - Message collisions in dense areas - Higher latency (especially with SDP) - Limited reception of low-power Class B signals
Vessel-Based AIS	- Good coverage along busy shipping lanes - Can leverage existing hardware on vessels - Bridges gaps between T-AIS and S-AIS - Scalable with fleet size	- Limited radio range (~20–40 nm) - Relies on vessel internet connectivity for data sharing - Continuous global coverage requires large fleet - No guaranteed continuous coverage in a single spot

In practice, this is done by connecting the AIS device's NMEA data output to a computer that transmits the data packets over an IP link (satellite, cellular, etc.). The data is then merged with other sources on shore to update global vessel tracks in real time, *MarineTraffic (n.d.)*.

To better illustrate the comparative strengths and limitations of different AIS modalities, Table I summarizes key advantages and disadvantages of terrestrial AIS, satellite AIS, and vessel-based AIS. This comparison helps clarify the complementary nature of these systems and supports the argument for integrated, multi-source AIS strategies to achieve reliable and continuous maritime coverage.

2.4. Main AIS Data Providers and Market Consolidation

The AIS data market is currently dominated by a few key players. Key players are Kpler (after acquisitions of MarineTraffic, FleetMon, and Spire Maritime), S&P Global (after acquisition of Orbcomm's AIS services), Lloyd's List Intelligence, Vesseltracker, and Clarksons. Community-based networks such as AIS Hub and VesselFinder also contribute through crowdsourced terrestrial feeds. Overall, the market has consolidated into a small number of providers offering global maritime intelligence solutions.

2.5. Previous Work

AIS coverage has remained a key issue in literature. *Svanberg et al. (2019)*, after reviewing 189 AIS-related publications, emphasized the importance of developing alternative methods for AIS data collection. Among the proposed solutions, aircraft-based AIS detection has been introduced as a way to complement satellite and land-based systems.

Most existing studies on AIS performance have adopted satellite-focused perspectives, primarily due to the limited range of shipborne AIS receivers. As a result, the research has largely addressed satellite system limitations, improvement strategies and constellation design. In contrast, vessel-based coverage analysis has received little attention and no dedicated study focusing on this topic could be identified in the current literature. This paper aims to address that gap by proposing a vessel-based perspective on AIS coverage assessment.

To evaluate AIS tracking performance, several studies have employed grid-based spatial analysis. *Skauen (2015),* for instance, introduced a methodology based on a 2° × 2° grid framework to assess the probability of satellite re-detection of ships. By incorporating AIS message timestamps, the study quantified spatial tracking capabilities through comparisons between satellite access frequency and actual vessel detection events. The choice of grid cell size was highlighted as a practical trade-off between computational efficiency and spatial resolution.

Following a similar approach, *Hasbi and Kamirul (2020)* applied the same grid-based method to evaluate the system and receiver performance of LAPAN-A2 and LAPAN-A3 satellites, which operate in equatorial and polar orbits. These findings reinforce the effectiveness of grid-based analysis for measuring AIS system performance. Building on this foundation, this paper applies grid-based spatial analysis to evaluate vessel-based AIS coverage.

Beyond satellite-based solutions, *Plass et al. (2015)* explored an alternative perspective by investigating the use of aircraft to detect AIS signals. Their study showed that aircraft operating at altitudes between 8,500 and 10,000 m can cover a radius of ~370 km. Due to their lower altitude and smaller footprint, aircraft experience less signal collision compared to satellites. The study concluded that airliner-based AIS detection offers a promising, cost-effective complement to existing terrestrial and satellite-based systems, particularly for filling coverage gaps.

In addition to coverage-related studies, this paper also considers AIS-based traffic analysis. Specifically, it focuses on extracting traffic routes and densities using annual AIS data. In the litera-

ture, *Wang et al. (2019)* proposed a method for deriving Global Shipping Networks (GSNs) from historical AIS data using DBSCAN clustering. Their approach identifies stop events, such as terminal visits and links them to specific ports and countries, allowing for a multi-layered representation of maritime traffic.

Similarly, *Lei et al. (2016)* introduced the Maritime Traffic Route Discovery (MTRD) framework to detect common vessel movement patterns. Their methodology involves identifying frequent movement regions and trajectory trends to produce generalized traffic routes.

Further contributions to AIS system research include efforts to optimize coverage through intelligent coordination of assets. *De Cubber et al. (2022)*, for example, presented a distributed optimization framework for unmanned maritime systems (UMS). Their approach focuses on efficient patrolling of maritime areas while reducing the risk of vessel instability under varying sea conditions. The framework integrates behavior-based control, offline learning of movement strategies and real-time adaptation based on wave height, visibility and inter-agent spacing.

In summary, although significant progress has been made in AIS data analysis and system optimization, the literature lacks a focused study on vessel-based AIS coverage. This paper addresses this gap by applying spatial analysis methods to evaluate AIS reception from a shipborne perspective.

3. Research material and methods

The foundation of this study lies in exploring the potential and performance of vessel-based AIS stations in delivering continuous maritime coverage across the globe. As outlined in the introduction section, this research is driven by the following pivotal questions: (1) Can vessel-based AIS stations provide adequate coverage of global vessel traffic? (2) How many vessels equipped with AIS receivers are necessary to achieve comprehensive and efficient coverage? (3) Does coverage effectiveness decrease as fleet size increases and at what point does adding more vessels result in diminishing returns?

3.1. Theoretical Framework

At the core of the framework is the assumption that vessels equipped with AIS receivers act as mobile monitoring stations. These station vessels are assumed to detect AIS signals broadcast by nearby vessels referred to as target vessels within a defined coverage radius. The presence of a station vessel in a given region during the same time interval as a target vessel implies potential signal reception and thus constitutes coverage.

To operationalize this concept, the study introduces a grid-based spatial model and time-segmented simulation logic. Each simulation cycle considers the real-time spatial distribution of vessels, their location within predefined tiles and their activity at 1-minute time intervals. Using both raw AIS data and aggregated summaries, the study evaluates how different fleet sizes affect detection rates and spatial distribution of covered regions.

Furthermore, the simulation examines how coverage scales with the number of station vessels, seeking to identify the point at which adding additional vessels yields diminishing returns. This analysis is grounded in the principle of diminishing marginal utility, where each additional station vessel contributes progressively less to overall global coverage after a threshold is reached.

The framework also introduces the concept of continuous coverage by shifting from binary vessel detection to hour-by-hour position capture. This approach allows for assessment of both spatial completeness and temporal persistence—key attributes for ensuring that vessel-based AIS can support maritime monitoring in a reliable and scalable manner.

3.2. Definitions & Assumptions

To ensure clarity and consistency throughout the study, several critical terms and assumptions are defined below:

- IMO Number: The IMO number is a unique, permanent identifier assigned to ships by the International Maritime Organization. Composed of the prefix "IMO" followed by seven digits, this number remains unchanged over a vessel's lifetime, regardless of changes in its flag, ownership, or name. IMO numbers are assigned to a wide range of vessels including cargo ships (≥ 300 GT), passenger ships (≥ 100 GT), high-speed craft, mobile offshore units and certain fishing vessels exceeding 12 m in length, *IMO (n.D.)*.
- Station Vessel: Refers to cargo and tanker vessels that are theoretically equipped with AIS receivers. These vessels are hypothesized to collect AIS signals from nearby vessels as they traverse global waters, functioning as mobile monitoring platforms.
- Target Vessel: Any vessel transmitting AIS data, identified by a valid IMO number, which can potentially be detected by the station fleet. Target vessels include all categories and types, regardless of operational area or purpose.
- Tile: A square-shaped fixed spatial grid unit used to segment the Earth's surface for analysis. The globe is partitioned into uniform tiles to facilitate spatial organization and regional classification of vessel movement.

Key Assumptions:
- All AIS-equipped vessels operate continuously without data transmission interruptions.
- AIS message reception by station vessels is assumed to be perfect, i.e., with no signal loss or interference.
- Vessels behave consistently with respect to their equipment and operational roles across the analysis period.
- Station vessels maintain a consistent coverage range (5×5 grid) (explained in section 3.3.1).

3.3. Spatial & Temporal Framework

3.3.1. Tiles

For this study, 1024 x 1024 spatial indexing is employed, dividing the world into a uniform grid of cells, each identified by an (x, y) coordinate. This indexing method partitions the world into 1024 cells along the x-axis and 1024 cells along the y-axis, resulting in a total of 1,048,576 (1024 x 1024) grid cells. The grid is defined based on the geographical boundaries of the Earth, with longitude ranging from -180° to 180° and latitude ranging from -90° to 90°. Each grid cell represents a specific geographic area, with dimensions calculated as follows:

These grid cells vary in size depending on their location on the Earth's surface. The area of each cell ranges between 11.468 km² and 1,531.603 km², with an average of ~485.71 km². The side lengths of the cells range between 3.386 km and 39.136 km, with an average side length of 18.49 km and diagonal lengths range from 4.789 km to 55.346 km, averaging 26.15 km.

This spatial indexing system is fundamental for analysing the distribution of vessel activity within each grid cell, enabling detailed spatial analysis of AIS data over large geographical areas.

Tiles are combined with marine regions which are the intersect of the Exclusive Economic Zones and IHO areas taken from MarineRegions.org. Each tile is assigned to region, subregion and microregion based on the average latitude and longitude.

3.3.2. Temporal Resolution

To evaluate the effectiveness of vessel-based AIS stations, two complementary analyses are conducted. One focused on basic detection and the other on temporal continuity. The first analysis aims to determine the number of station vessels required to detect at least one position from each target vessel. In this context, a vessel is considered "covered" if it is detected even once during the day. The second analysis extends this by dividing each day into hourly intervals and calculating how many vessel–1-hour position combinations are captured as the fleet size increases. This metric is critical, as true continuity in monitoring depends on temporal density. A single position per vessel per day is insufficient and achieving at least hourly resolution is essential for reconstructing voyages and enabling meaningful operational insights.

3.4. Simulation Setup

3.4.1. Simulation Methodology

The simulation framework developed to evaluate real-time coverage potential based on raw, timestamped AIS data focuses on assessing vessel-based AIS coverage at a granular, 1-minute interval level. The key steps in the simulation process are as follows:

- ~10 days were randomly selected across different months of the year to ensure diversity in vessel behavior and account for seasonal changes. Each day was segmented into 1-minute intervals, enabling high-resolution analysis of vessel movement and presence.
- For each selected day, a population of cargo and tanker vessels with valid IMO numbers was identified. These vessel types were chosen because of their longer voyages and relevance for monitoring remote maritime areas. Random samples of varying sizes (e.g., 50, 200, 1000, 3000) were drawn from this pool to simulate different station fleet configurations.
- Each station vessel is assumed to have an AIS reception range of 20–40 nm (37–74 km). Given the average diagonal tile length of ~26.15 km, this range covers a 5×5 grid of tiles surrounding the vessel's current location. If a station vessel is present in a tile at a given minute, all vessels in that 5×5 area at that same minute are considered within its coverage.
- Target vessels which any IMO vessels broadcasting AIS signals are mapped to their respective tiles and timestamps and lookup tables are created. The simulation checks whether a station vessel was present in a 5×5 tile neighborhood during the same minute and if so, marks that target vessel's position as "covered."
- This process is repeated across multiple randomly chosen days and station fleet sizes. Each combination is run multiple times to account for variability in random sampling and the results are averaged to ensure robustness.

3.4.2. Example Illustration

To increase the clarity and transparency of the simulation process, we present a detailed example based on a real vessel included in the simulation: the container ship Gdansk Express with IMO number 9943877. This example illustrates how vessel-based AIS coverage is calculated through grid-based spatial logic and 1-minute temporal segmentation.

Step 1: Identifying Station Vessel Activity
In February, Gdansk Express traveled from Europe to Asia via South Africa. Its position can be approximated as Lat −28.99, Lon 40.44 at 24 February 2024, 00:00 UTC, and Lat −27.41, Lon 46.69 at 25 February 2024, 00:00 UTC. An excerpt of hourly approximate positions is provided in Table II, and the full path with coverage range is shown in Fig.3.

Table II: Excerpt of hourly approximate AIS positions of Gdansk Express

Timestamp	Latitude (°)	Longitude (°)
2025-02-24 00:00	-28.99	40.44
2025-02-24 03:00	-28.69	41.20
2025-02-24 06:00	-28.38	41.93
2025-02-24 09:00	-28.24	42.71
2025-02-24 12:00	-28.08	43.61
2025-02-24 15:00	-27.91	44.43
2025-02-24 18:00	-27.73	45.20
2025-02-24 21:00	-27.52	46.15
2025-02-25 00:00	-27.41	46.69

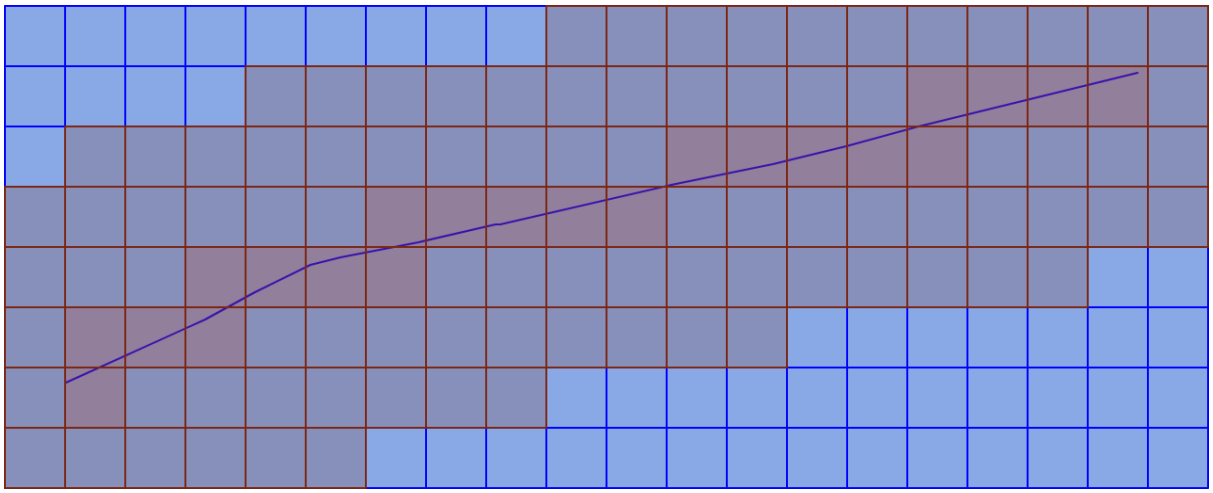


Fig.3: AIS positions of *Gdansk Express* on 24 February 2025, with the blue line showing its track, red tiles indicating visited areas, brown tiles within reach, and blue tiles out of reach.

Each message was assigned to a spatial cell and rounded down to the nearest 1-minute interval. This process created a list of (tile, time) combinations representing all the 1-minute intervals in which Gdansk Express could potentially receive AIS signals from nearby vessels. Each of these records represents a moment when Gdansk Express was theoretically acting as a mobile AIS station vessel with a 5×5 tile coverage area, centered around the tile it was located in at that specific minute.

Step 2: Building the Detection Map Using Raw AIS Data
For each (tile, minute) combination, the simulation expands the coverage region by looking at a 5×5 neighborhood of tiles centered on the given tile. This expansion mimics an AIS reception range of roughly 20–40 nm (based on grid tile size and diagonal distance).

A precomputed lookup table includes all IMO numbers of vessels that were broadcasting AIS messages within each tile's 5×5 neighborhood at every minute of the day. By joining Gdansk Express's tile-time pairs with this table, the simulation retrieves all vessels that were theoretically detectable. An excerpt of this matching is shown in Table III, where we display several examples of vessels that were co-located (within 5×5 tile region) and broadcast at the same time as Gdansk Express.

Table III: Excerpt of co-detected vessels via 5×5 tile neighborhood logic

Timestamp	GDANSK Tile (xy)	Detected IMO Vessels (sample)
2025-02-24 01:18:00	1006270597	9406520, 9563407, 9846079, 9943877
2025-02-24 03:00:00	1006290597	9334882, 9645451, 9786841, 9830094, 9943877, 9972452
2025-02-24 07:43:00	1006320595	9352391, 9668934, 9830094, 9939620, 9943877
…	…	…

This lookup enables the simulation to mark any of those vessels as “detected” by Gdansk Express at that minute, provided their AIS message timestamp and location match.

Step 3: Interpreting Coverage
The simulation considers any target vessel listed in the same tile-time region as a "covered vessel". For example, at 03:00 UTC, Gdansk Express was located at tile 1006290597. The 5×5 tile neighborhood of this tile included AIS messages from vessels such as:

- IMO 9334882
- IMO 9830094
- IMO 9645451
- IMO 9893993
- and others

This process is repeated for each minute of Gdansk Express's movement and applied to every station vessel in the simulation sample. The cumulative result allows for a quantifiable measure of coverage: how many distinct IMO-registered vessels were detected (once per day or once per hour), and how many unique vessel–hour combinations were captured. Based on the simulation logic, for each of these moments, a 5×5 neighborhood around the tile was evaluated for other vessels transmitting AIS at the same minute.

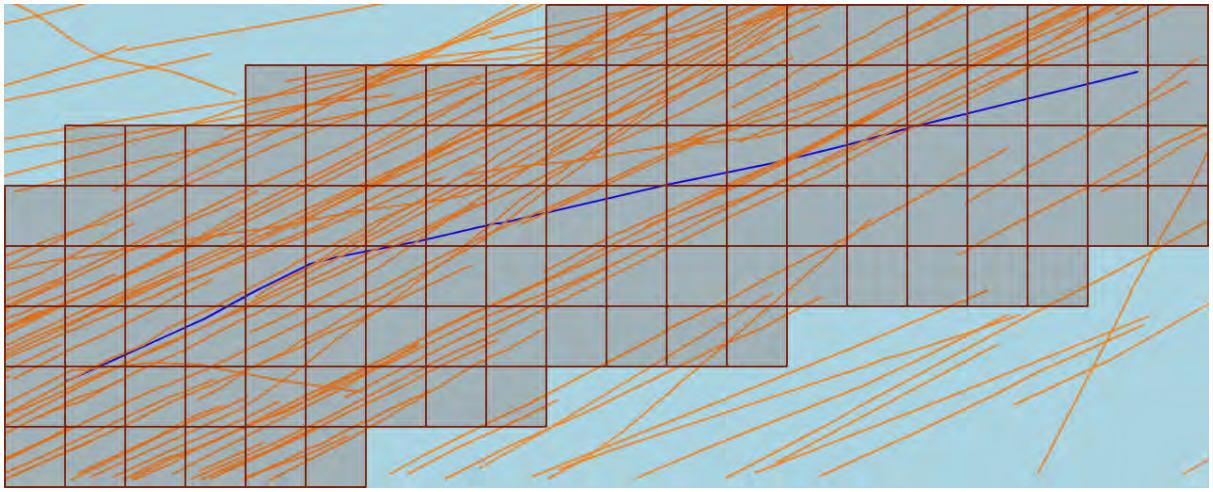


Fig.4: Voyage of Gdansk Express (IMO 9943877) is shown in blue. Orange lines represent the voyages of target vessels, with segments inside the red grids indicating targets detected within its 5 × 5 tile coverage. The red grids denote the areas covered by Gdansk Express.

From this, Gdansk Express was found to have:

- Detected 79 distinct IMO-registered target vessels
- Generated 129 total detection events (some vessels were detected multiple times in different tiles/minutes)
- Detected vessels across 30+ distinct grid areas, spanning different longitudes and latitudes, indicating good spatial diversity
- Detected up to 8 vessels simultaneously during peak coverage minutes (e.g., at 03:00 UTC)

This concrete example demonstrates how a single vessel, moving with typical ocean-going dynamics, can act as a distributed AIS sensor. Gdansk Express alone accounted for 79 unique vessel detections in a single day, providing both spatial and temporal coverage across its journey. When scaled to hundreds or thousands of such vessels, the network effect becomes powerful which enables large-scale tracking coverage even in remote waters where terrestrial or satellite solutions may face limitations.

3.4.3. Example Walkthrough: 100 Station Vessels on 2025-02-24

To evaluate performance at a more realistic fleet scale, a simulation was conducted using 100 randomly selected vessels as mobile AIS stations. These vessels were selected from the full AIS dataset for 24.2.2025, filtered to include only terrestrial and satellite sourced messages. The goal was to examine the coverage performance when stations are deployed globally and act simultaneously.

Each station vessel's AIS data was used to generate a list of tile–time intervals it covered. These were then matched against a precomputed lookup table that contained all vessels observed in each tile and 5×5 neighboring grid within a 1-minute resolution. From this, a comprehensive list of detected vessels and their respective AIS records was extracted.

Step 1: Selection of Station Vessels - 100 IMO numbers were sampled at random. These vessels act as mobile AIS observation stations. Table IV shows an excerpt of the metadata for selected vessels, including name, type, size, and flag.

Table IV: Excerpt of selected sample station vessels

imo	shipname	shiptype	type_name	length	width	flag
9591014	JIN HAI ZHONG	70	Bulk Carrier	190.0	32.3	CN
9477505	CHEM TAURUS	80	Oil/Chem. Tanker	145.5	23.7	LR
9610432	DIVINE ACE	70	Vehicles Carrier	200.0	32.2	PA
8822442	USNS GUADALUPE	89	Repl. Vessel	206.5	29.7	US
8138815	ISTANBUL BUNKER	80	Bunkering Tanker	64.6	9.4	TR
…	…	…	…	…	…	…

Step 2: AIS Data of Station Vessels - All AIS positions reported by these 100 vessels on the same day were retrieved from the AIS dataset. Each message was mapped to a spatial tile (xy) and truncated timestamp (time_1min).

Table V: Excerpt of AIS positions of example station vessels

IMO	Latitude	Longitude	Timestamp	Tile (xy)
9591014	32.219414	119.215900	2025-02-24 00:00:41	1008510415
9477505	37.675940	9.472756	2025-02-24 00:37:31	1005380396
9610432	-32.569073	164.655940	2025-02-24 00:00:30	1009800610
8822442	21.364475	-157.942410	2025-02-24 00:28:15	1000620449
8138815	40.775127	29.628174	2025-02-24 06:00:36	1005960384
…	…	…	…	…

Step 3: Lookup Table for Station Tile - To determine which vessels are potentially observable by a station vessel at a given time, the simulation uses a precomputed lookup table. This table lists all IMO numbers present in a 5×5 tile neighborhood for each minute. The following example shows entries for tile 1008510415, which was visited by station vessel JIN HAI ZHONG (IMO: 9591014) at 2025-02-24 00:00:41.

Table VI: Excerpt of lookup table for station tile

Tile (xy)	Time (UTC)	Detected IMO List (excerpt)	Count
1008510415	2025-02-24 00:00:00	{9591014, 9983968, 9572551, 9262998, ...}	25
1008510415	2025-02-24 00:06:00	{9481893, 9490698, 9816593, 9928889, ...}	26
1008510415	2025-02-24 00:13:00	{9129603, 9572551, 9591002, 9988970, ...}	18
1008510415	2025-02-24 00:17:00	{9536014, 9563251, 9607825, 9622057, ...}	27
…	…	…	…

Each row shows how many distinct IMO numbers were present in the station's local tile neighborhood at each minute, forming the foundation for target detection.

Table VII: Coverage summary of 100 randomly selected station vessels on 24.2.2025

Metric	Value
Number of Station Vessels	100
AIS Records from Station Vessels	16,393
Total Tiles Covered by Station Vessels	575
Active 1-Minute Intervals	888
Total Tile-Time Combinations	16,110
Detected Target Vessels (Unique IMOs)	14,898
Detected AIS Records from Target Vessels	940,084

The simulation using 100 randomly selected station vessels on 24.2.2025 produced a total of 16,393 AIS records from the stations. These vessels covered 575 unique spatial tiles across 888 one-minute intervals, resulting in 16,110 distinct tile–time combinations where target detection was possible. Within this coverage, the simulation identified 14,898 unique target vessels (based on IMO numbers). The AIS data for these targets amounted to 940,084 records, which were retrieved for further analysis. This output serves as a representative example of the detection capacity and data volume generated in one simulation iteration.

Fig.5: Global distribution of AIS detections by 100 randomly selected station vessels on 24.2.2025. Each red line represents vessel movement, and grey points indicate AIS messages detected from surrounding target vessels within each station vessel's 5×5 tile coverage.

3.4.4. Validation Strategy

A predetermined fleet of actual vessel-based AIS stations was used in a series of tests to confirm the simulation results' accuracy and realism. This known fleet was tested by the simulation framework with different temporal resolutions (1-, 5-, and 15-minute intervals) and different coverage assumptions, such as 1×1, 3×3, and 5×5 tile grids. The outcomes of these configurations were compared with the real fleet's coverage. The 5×5 grid with 1-minute interval granularity generated the most accurate and representative results, according to both mathematical analysis and empirical comparisons. In particular, this arrangement closely matched the observed behavior of the actual station fleet in terms of both the quantity of detected target vessels and their spatial distribution across

tiles. Based on this validation process, the 5×5 spatial assumption and 1-minute temporal segmentation were adopted as the standard setup for all subsequent simulation iterations.

3.4.5. Computational Setup

A PostgreSQL database was used to store and manage the AIS data used in this study, and DBeaver served as the main interface for data exploration and querying. Python was used to implement the simulation calculations, data aggregations, and processing logic, with Visual Studio Code serving as the development environment. Every analysis was carried out locally on a desktop computer.

4. Results

In this study, as highlighted in methodology section in detail, for around 10 specific days selected randomly between 2024 January – 2025 March, different fleet sizes ranging from 1,000 to 15,000 with 1,000 increments were simulated. For each day and each fleet size simulation were running 10 times. That makes over 1,500 simulations handled. For each fleet size, output which is total number of distinct IMO target vessels were averaged and standard deviations were calculated. The same thing done for hourly coverage for same days, same fleet sizes. But the grouping at the end was not for day, it was for hours in a day.

Table VIII: Structure and scale of the AIS data used in the analysis

Date	Total AIS Messages	Unique IMOs	Unique Tiles	Avg. Positions per IMO
2025-03-24	27,349,649	65,604	57,286	416.9
2025-03-17	23,655,368	64,953	57,655	364.2
2025-03-10	21,611,810	65,013	54,676	332.4
2025-03-02	22,455,815	62,797	55,452	357.6
2025-02-24	25,132,746	63,572	55,789	395.3
2025-02-17	22,790,827	63,875	55,357	356.8
2024-12-15	23,021,753	58,925	42,731	390.7
2024-10-15	7,574,746	61,224	39,491	123.7
2024-07-15	19,742,520	64,631	51,324	305.5
2024-04-15	8,435,863	62,980	48,015	133.9
2024-01-15	7,345,856	67,355	91,021	109.1

To provide context for the simulation results, Table VIII summarizes the structure and scale of the AIS data used in the analysis. Each dataset corresponds to a full day of AIS messages filtered by terrestrial and satellite sources. On average, each selected day contains over 20 million AIS messages from approximately 60,000–67,000 unique vessels (IMO numbers). The number of unique spatial tiles (xy) per day ranges from ~39,000 to over 91,000, reflecting the geographical spread of maritime activity. Average AIS message count per vessel also varies by day, with peak reporting activity exceeding 400 positions per IMO, and quieter days (e.g., 2024-10-15). For selected days, the full set of terrestrial and satellite AIS messages was retrieved and processed at 1-minute granularity.

Table IX: Key Dataset Characteristics (Averages per Day)

Metric	Value
Unique IMO-registered vessels	~63,721 (±2,163)
Total AIS messages	~19.0 million (±7.1 million)
Active spatial tiles (1024x1024 grid)	~55,345 (±12,660)
Avg. AIS positions per vessel	~299 (±112)
Share of satellite messages	~1.44% (±2.94%)

Each AIS record was mapped to a spatial tile and truncated to 1-minute resolution. The simulation framework assumes a 5×5 tile reception radius (~20–40 nm) for each station vessel and performs minute-level matching between station and target vessels. For each fleet size (from 500 to 15,000), 10 independent simulations were run per day, totaling over 1,500 iterations.

4.1. Daily Detection Coverage

The first analysis assesses how effectively vessel-based AIS stations can detect other vessels at least once during a given day. Here, coverage is defined as the presence of at least one AIS-detecting station vessel within reception range of a target vessel at any time during the day.

For each subset of station vessels, distinct 1-minute time interval and tile combinations were analyzed. The underlying assumption is that if a station vessel is present in a tile during a specific 1-minute interval, it is capable of receiving AIS positions from target vessels located within that tile and its surrounding 2-tile radius which forms a 5×5 coverage grid. Any target vessel within this neighborhood at the same minute is considered covered. This setup provides a robust dataset for evaluating the effect of increasing station fleet size on global AIS coverage.

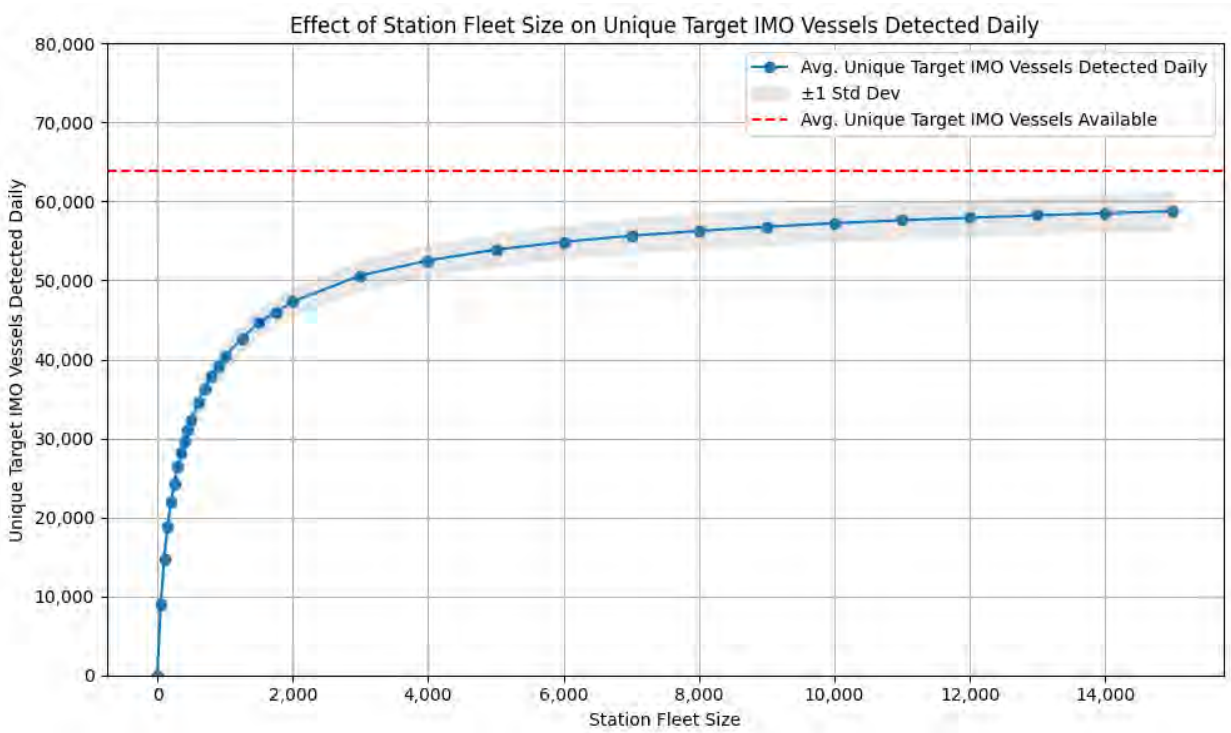


Fig.6: Effect of station fleet size on unique target IMO vessels detected daily

Table X: Target IMO vessels detected daily and % world fleet

Fleet Size	Detected Vessels (± Std Dev)	% of ~63,000
500	32,264 (± 1,347)	51.20%
1,000	40,423 (± 1,546)	64.20%
2,000	47,322 (± 1,671)	75.10%
3,000	50,622 (± 1,821)	80.30%
4,000	52,525 (± 1,901)	83.40%
5,000	53,901 (± 1,999)	85.60%
10,000	57,241 (± 2,206)	90.90%
15,000	58,781 (± 2,296)	93.30%

On average, the total number of distinct IMO-registered vessels present per day was approximately 63,000. This represents the upper limit of coverage achievable through station vessels on a given day. As illustrated in Fig.6, coverage increases sharply with initial fleet growth. The first 50 station vessels detect approximately 9,400 unique target vessels. Expanding the fleet to 100 vessels raises this figure to around 15,000. A fleet of 200 station vessels is capable of covering more than 22,000 distinct target vessels, while 400 stations can detect approximately half of the daily global fleet.

A fleet of 1,000 station vessels provides coverage for around 40,000 target vessels and 3,000 stations raise this to ~50,000. Notably, increasing the fleet from 3,000 to 10,000 adds coverage for only about 7,000 additional vessels, highlighting the rapid decline in marginal utility. In contrast, the initial 50 stations alone contributed coverage for over 9,000 vessels.

The analysis clearly shows that beyond a fleet size of 2,000–2,500, the marginal gains in coverage begin to taper off significantly. This inflection point marks the onset of diminishing returns, where further increases in fleet size yield limited additional coverage.

The analysis so far focuses on whether each vessel is detected at least once per day. However, continuous monitoring requires more granular temporal resolution. If the objective shifts to capturing vessel positions every hour, the number of potential combinations increases substantially reaching approximately 63,000 vessels × 24 h. This scenario is analyzed in the next section.

4.2. Hourly Detection Coverage

While the previous section focused on whether each target vessel was detected at least once per day, this section evaluates a stricter criterion which is detecting vessel activity at an hourly resolution. In this context, “vessel activity” is defined as each successful detection of a vessel in a given 1-hour window. With approximately 63,000 unique IMO vessels available on average per day, the theoretical maximum number of vessel–hour combinations are around 1.5 million (63,000 vessels × 24 hours).

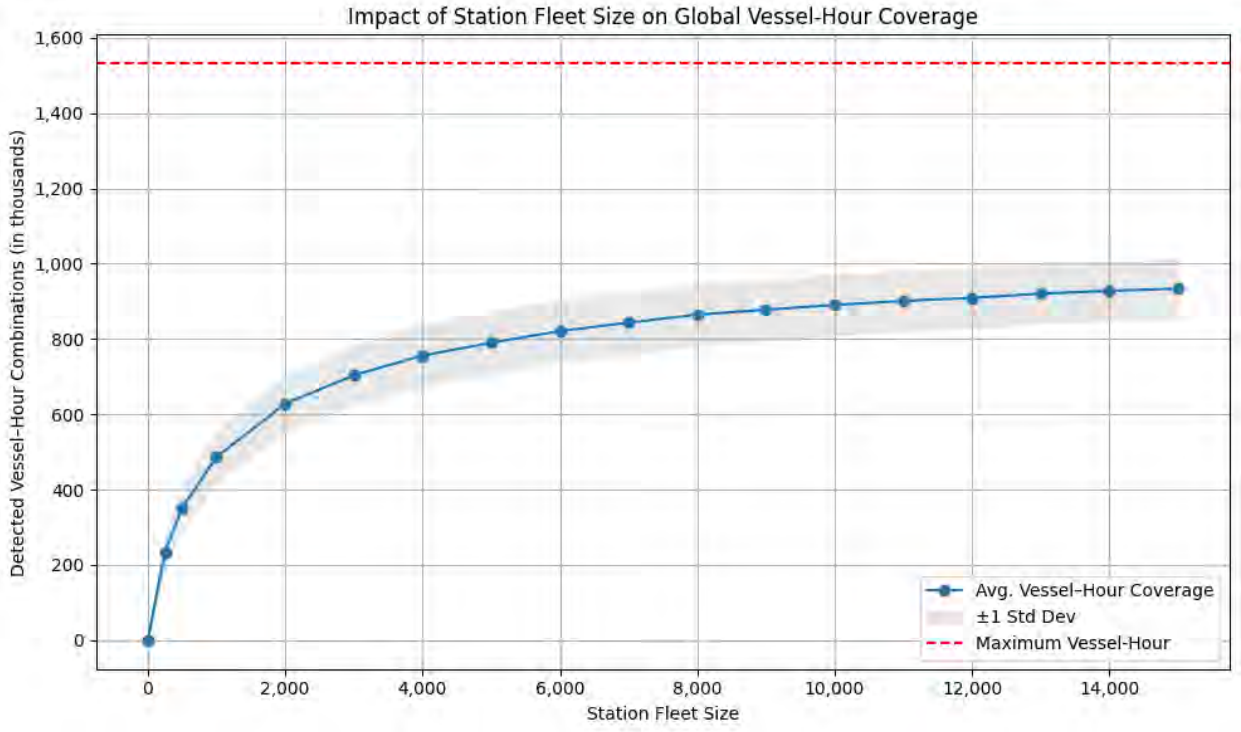


Fig.7: Impact of station fleet size on global vessel-hour coverage

Fig.7 illustrates the relationship between station fleet size and the number of vessel–hour activity records detected. As with daily detection, the trend demonstrates a steep increase at the lower end of

the fleet size spectrum. A fleet of just 200 station vessels captures around 400,000 vessel–hour combinations. Expanding the fleet to 1,000 stations increases coverage to roughly 700,000 combinations—less than half of the maximum possible activity.

Coverage continues to grow with fleet size, reaching about 950,000 vessel–hour records with 15,000 stations. However, the marginal gains diminish substantially beyond 3,000–4,000 vessels. For instance, expanding the fleet from 3,000 to 10,000 adds only ~150,000 additional vessel–hour detections. This plateau reflects the same saturation dynamics observed in daily detection analysis, but with more pronounced gaps due to the tighter temporal requirement.

The standard deviation bands in the figure highlight moderate variability in coverage depending on the sampled day and fleet composition. Nonetheless, the general trend is consistent: while vessel-based AIS coverage can scale significantly with fleet size, achieving full global hourly coverage would require a disproportionately large fleet. This result reinforces the challenge of achieving continuous temporal coverage using mobile AIS receivers alone. Even at high fleet sizes, a considerable fraction of hourly vessel activity remains undetected underscoring the need for either targeted deployment strategies or complementary use of terrestrial and satellite AIS sources.

4.3. Fleet Size Trade-offs and Scalability

The combined results offer valuable insights into the trade-offs involved in scaling vessel-based AIS deployment. At the lower end of the spectrum, an initial deployment of ~500 to 1,000 vessels is sufficient to capture significant portions of high-traffic maritime corridors. This level of coverage presents a cost-effective and practical entry point, particularly for hybrid monitoring strategies that aim to complement existing terrestrial and satellite AIS systems.

A moderate-scale deployment, involving around 2,000 to 3,000 station vessels, enables strong global daily detection capabilities. This fleet size is well-suited for applications that require consistent situational awareness and operational monitoring without the need for high-frequency updates. It offers a balanced solution between performance and resource requirements.

For applications demanding higher temporal fidelity, such as near-continuous tracking or voyage reconstruction, a larger fleet of 5,000 to 6,000 vessels or more is needed. While such deployments can provide partial hourly resolution and enhance detection consistency, they also exhibit clear signs of diminishing returns. Beyond a certain point, adding more vessels leads to only marginal gains in additional detections.

This scalability highlights the strategic value of vessel-based AIS as a complementary layer within the broader AIS infrastructure ecosystem. It is particularly beneficial in commercial or regulatory contexts where full temporal resolution is not always necessary, but wide geographic coverage and redundancy are essential.

5. Discussion

This study demonstrates the feasibility of vessel-based AIS stations as a scalable, mobile alternative for maritime traffic monitoring. The simulation results indicate that vessel-based AIS can achieve substantial global coverage, particularly at the daily resolution level, with a fleet of 2,000–3,000 equipped vessels. However, achieving near-continuous hourly resolution requires significantly more resources upwards of 6,000 vessels.

These findings confirm the hypothesis that marginal coverage gains decrease as fleet size expands, a dynamic consistent with the principle of diminishing marginal utility. While initial growth in station vessel numbers yields significant detection increases, beyond the 3,000-vessel threshold, additional deployments contribute incrementally to global coverage.

Comparing this vessel-based approach to satellite and terrestrial AIS confirms its complementary potential in global maritime surveillance. Satellite AIS excels in oceanic regions where land-based infrastructure is absent, offering global reach through constellations of LEO satellites. However, satellite AIS is hindered by several technical limitations as mentioned in section 2.2.1. Chief among these are message collisions, which arise when multiple ships using SOTDMA transmit simultaneously without coordination across large coverage footprints. A single satellite at 650 km altitude can observe areas with diameters up to 5,000 km, encompassing thousands of ships from distinct SOTDMA regions that cannot synchronize their slots. This leads to severe packet collisions, particularly in congested areas, as demonstrated by Cervera et al. (2010). Their findings show that when ship counts exceed 2,500, successful message reception drops dramatically, with the AIS channel nearing its limit of 4,500 messages per minute. Even with advanced processing techniques such as Spectrum Decollision Processing, there are still some problems ongoing such as latency coming from ground operations, *IALA (2011)*.

Moreover, terrestrial AIS, while offering high message fidelity and low latency, is fundamentally constrained by line-of-sight limitations and geography. Its typical range, limited to 15–40 nm from shore, makes it effective only in coastal and port areas, *MarineTraffic (2021)*. This leaves vast mid-ocean zones and sparsely monitored regions beyond the reach of fixed infrastructure.

In contrast, vessel-based AIS leverages ships already equipped with AIS transceivers to dynamically collect signals from nearby vessels. This mobility allows shipborne stations to fill in systemic coverage gaps in both S-AIS and T-AIS networks. These mobile receivers are especially effective in mid-ocean corridors and under-monitored developing regions, where neither terrestrial antennas nor satellite revisit frequency can guarantee consistent performance. The key operational challenge becomes maintaining reliable connectivity for data forwarding, particularly via satellite internet, although connectivity costs are expected to decline further with advances in maritime communications.

To date, no prior studies have simulated vessel-based AIS coverage on a global scale, marking this work as a novel contribution to the field. Most existing research has focused on satellite AIS, addressing its technical limitations, signal collision challenges, and constellation optimization. The only notable exception is the work by *Plass et al. (2015)*, which explored aircraft-based AIS detection as an alternative data collection method. Their findings demonstrated the feasibility of using airliners to bridge coverage gaps between coastal and satellite-based systems. Furthermore, *Svanberg et al. (2019)*, in their review of 189 AIS-related publications, underscored the continued importance of developing alternative AIS data collection strategies. While advancements in satellite technology and market competition may reduce bandwidth costs over time, the strategic value of complementary solutions such as vessel-based AIS remains highly relevant. This study fills a critical gap in the literature by demonstrating that vessel-based AIS can serve as a viable and scalable layer in the global maritime surveillance ecosystem.

Additionally, the practical feasibility of this approach is supported by trends in AIS data infrastructure. As described in section 2.4, AIS data markets are undergoing significant consolidation. Companies like Kpler now control both satellite and terrestrial data pipelines via acquisitions of MarineTraffic, FleetMon, and Spire Maritime. This trend could limit access or drive-up prices, especially for satellite data, making alternative or supplementary data sources like vessel-based AIS strategically valuable for both commercial and public-sector stakeholders. This study shows that vessel-based AIS is a good alternative source for global maritime surveillance and can help maintain a more balanced data ecosystem. By offering a decentralized and scalable coverage method, it can serve as a counterweight to satellite-based data monopolies and can potentially place downward pressure on pricing. For data buyers, this diversification enhances resilience and cost-effectiveness in accessing maritime information.

Despite the robust simulation setup, several limitations must be acknowledged. Several simplifying assumptions underlie the simulation framework. First, it is assumed that all station vessels are continuously operational and maintain uninterrupted AIS reception capability throughout the day. In

reality, signal loss may occur due to equipment issues, vessel downtime, or communication interruptions. It is also assumed that each station vessel can always cover a 5×5 tile area, although this may vary depending on the actual reception radius and local environmental conditions. Furthermore, tile sizes vary across the globe due to the Earth's curvature, which may affect the effective coverage area per vessel depending on geographic location. Weather and terrain may also influence AIS signal propagation, but these factors are not modelled in this simulation. The AIS data used in this study may not capture all vessels globally. The quality and completeness of simulation outputs depend directly on the comprehensiveness of the input dataset, particularly the positional records of station vessels. Additionally, randomly selecting station vessels for each simulation iteration may not reflect the operational reality of commercial or government-owned fleets, which often consist of vessels of specific types concentrated in particular regions. Therefore, while randomization supports generalizability, it may not represent the performance of actual deployed fleets with known biases or strategic objectives.

Nonetheless, the approach demonstrates high adaptability. It provides a strong foundation for strategic AIS deployment planning, whether for commercial shipping analytics, regulatory monitoring or global maritime situational awareness.

6. Summary/Conclusions

This paper evaluated the feasibility of vessel-based AIS stations in terms of continuous maritime traffic monitoring. Through the simulation designed, it examined how global AIS coverage is affected by different fleet sizes both at daily and hourly resolutions. The study investigated three central questions: (1) Can vessel-based AIS provide good enough global maritime coverage? (2) How many station vessels are needed for sufficient coverage? (3) Does increasing fleet size yield diminishing returns in terms of coverage?

To answer these questions, a high-resolution simulation framework was developed using real-world AIS data collected over eleven days between January 2024 and March 2025. The data included over 20 million AIS messages per day, representing more than 63,000 unique IMO-registered vessels across diverse maritime regions and seasons. Spatial segmentation was conducted using a 1024×1024 grid, and vessel positions were analyzed at 1-minute intervals to capture both spatial and temporal aspects of AIS coverage.

The simulation evaluated a range of fleet sizes from 500 to 15,000 hypothetical station vessels using randomized sampling, multiple iterations, and comprehensive detection logic. Coverage was assessed in two ways: (1) daily detection, which required each vessel to be detected at least once per day, and (2) hourly detection, which aimed to observe each vessel within every 1-hour window.

Results show that vessel-based AIS stations can provide adequate daily coverage of global vessel traffic both in daily and hourly resolutions. With around 2,000-2,500 vessels, over 75% of the global IMO-registered fleet can be detected at least once daily. For continuous detection, which is in hourly resolution, significantly larger fleet of 5,000-6,000 vessels is required. Coverage effectiveness decreases as fleet size increases.

The study highlights the potential of vessel-based AIS as a complementary source in maritime surveillance, especially in bridging coverage gaps left by terrestrial and satellite systems and offers vessel-based AIS as alternative source in the AIS data market which currently experiences consolidation.

Future research can extend this simulation model. A cost-benefit analysis of various deployment strategies, including the trade-offs between fleet size, coverage quality, and communication infrastructure requirements, would provide valuable insight for both commercial and regulatory stakeholders. Future studies could also explore what specific investments are necessary to scale up vessel-based AIS coverage and identify the operational and logistical challenges involved in

equipping, maintaining, and coordinating a distributed fleet of AIS-receiving vessels. Finally, the simulation framework developed in this study can be adapted to evaluate satellite and terrestrial AIS coverage under similar spatial and temporal conditions, allowing for direct comparisons between these complementary technologies and enabling more informed decisions on hybrid AIS infrastructure planning.

References

CERVERA, M.A.; GINESI, A.; ECKSTEIN, K. (2010), *Satellite-based vessel Automatic Identification System: A feasibility and performance analysis*, Int. J. Satellite Communications and Networking 29/2, pp.117–142

CHEN, Y. (2014), *Satellite-based AIS and its Comparison with LRIT*, TransNav Int. J. Marine Navigation and Safety of Sea Transportation 8/2, pp.183–187

FOURNIER, M.; HILLIARD, R.C.; REZAEE, S.; PELOT, R. (2018), *Past, present and future of the satellite-based automatic identification system: areas of applications (2004–2016)*, WMU J. Maritime Affairs 17/3, pp.311–345

HASBI, W.; KAMIRUL, N. (2020), *Tracking capability and detection probability assessment of Space-Based Automatic Identification System (AIS) from equatorial and polar orbiting satellites constellation*, IEEE Access 8, pp.184120–184136,

IALA (2016), *An overview of AIS (G1082, Edition 2.1)*, https://www.iala-aism.org

IALA (2021), *The use of the automatic identification system (AIS) in marine aids to navigation services (R0126 (A-126), Edition 2.0)*, https://www.iala.int/product/r0126/

IMO (2015), *Revised guidelines for the onboard operational use of shipborne automatic identification systems (AIS) (Resolution A.1106(29))*, Int. Mar. Org., London, https://wwwcdn.imo.org/localresources/en/OurWork/Safety/Documents/AIS/Resolution%20A.1106(29).pdf

KPLER (2023a), *Kpler acquires MarineTraffic and FleetMon for maritime sector expansion*, https://www.kpler.com/blog/kpler-acquires-marinetraffic-and-fleetmon-for-maritime-sector-expansion

KPLER (2023b), *Kpler acquires Spire Maritime to enhance maritime data and analytics capabilities*, https://www.kpler.com/blog/kpler-acquires-spire-maritime-to-enhance-maritime-data-and-analytics-capabilities

LEI, P.; TSAI, T.; PENG, W. (2016), *Discovering Maritime Traffic Route from AIS network*, 18th Asia-Pacific Network Operations and Management Symposium (APNOMS)

MARINETRAFFIC (2021), *Global ship tracking intelligence: Understanding AIS: Terrestrial vs satellite AIS tracking*, https://cdn2.hubspot.net/hubfs/2655099/Resources/Understanding%20Satellite%20AIS%20Tracking%20-%20MarineTraffic%20Resources.pdf

PLASS, S.; POEHLMANN, R.; HERMENIER, R.; DAMMANN, A. (2015), *Global Maritime Surveillance by Airliner-Based AIS Detection: Preliminary analysis*, J. Navigation 68/6, pp.1195–1209

ROBARDS, M.; SILBER, G.; ADAMS, J.; ARROYO, J.; LORENZINI, D.; SCHWEHR, K.; AMOS, J. (2016), *Conservation science and policy applications of the marine vessel Automatic Identification System (AIS)—a review,* Bulletin of Marine Science, 92(1), pp.75–103

SKAUEN, A.N. (2015), *Quantifying the tracking capability of space-based AIS systems*, Advances in

Space Research 57/2, pp.527–542

SVANBERG, M.; SANTÉN, V.; HÖRTEBORN, A.; HOLM, H.; FINNSGÅRD, C. (2019), *AIS in maritime research*, Marine Policy 106, 103520

UNCTAD (2024), *Review of maritime transport 2024: Navigating maritime chokepoints (UNCTAD/RMT/2024)*, United Nations, https://shop.un.org

YANG, D.; WU, L.; WANG, S.; JIA, H.; LI, K.X. (2019), *How big data enriches maritime research – a critical review of Automatic Identification System (AIS) data applications*, Transport Reviews 39/6, pp.755–773

Constructing Design Information Exchange Relations in Shipbuilding Using Large Language Models

Kohei Matsuo, National Maritime Research Institute, Tokyo/Japan, kohei@m.mpat.go.jp

Abstract

This paper explores the use of large language models (LLMs) for constructing fine-grained design information exchange relations in ship design. While traditional drawing-level management obscures dependencies and slows design processes, our approach extracts design information units from heterogeneous sources, classifies them, and infers exchange relations using LLMs. A simplified case study of engine room design illustrates feasibility. The study clarifies potential benefits for traceability and automation, while identifying key challenges in multimodal information extraction, granularity control, and relation inference that must be addressed for practical deployment.

1. Introduction

Ship design has traditionally been managed at the level of drawings. Each drawing functions as a composite representation that encompasses multiple design elements—dimensions, materials, arrangements, and process instructions—integrated into a holistic view of the product. This drawing-centered paradigm arose for several reasons: designers needed to grasp the overall structure of the vessel at once; drawings naturally combined heterogeneous design information in a single artifact; and until recently, there were few practical means to systematically manage design information at finer granularity. Attempts to fragment drawings into smaller units often risked losing the contextual relationships necessary for understanding the whole.

While this drawing-based approach has long served the needs of the shipbuilding industry, it also introduces several challenges. First, the granularity of information exchange is coarse, leading to delays: downstream engineers often must wait until a drawing is finalized before accessing information that could have been utilized earlier. Second, the relationships among pieces of design information remain largely tacit and person-dependent, making them difficult to trace, reproduce, or automate. Third, design management itself is opaque, as the flow of information between drawings is neither explicitly recorded nor visualized. These limitations contribute to extended lead times, hinder knowledge reuse, and obstruct the application of digital automation in design workflows.

In the past, such limitations were tolerable because ship design projects could still be managed effectively at the drawing level. However, in the face of increasing product complexity, tighter delivery schedules, and greater integration across disciplines, the drawing-centric paradigm has become insufficient. Furthermore, modern Product Lifecycle Management (PLM) environments now provide infrastructures that make it technically feasible to manage design information at much finer granularity.

This motivates a shift toward design management at the level of design information units rather than entire drawings. Yet, extracting such fine-grained design information and systematically linking them into coherent relationships remains a formidable task. Manual efforts or rule-based methods have proven inadequate in coping with the diversity, unstructured nature, and scale of ship design data.

Recent advances in large language models (LLMs) offer a potential breakthrough. With their contextual understanding and reasoning capabilities, LLMs can be leveraged to extract design elements from heterogeneous sources (drawings, specifications, memos, and communications) and to infer the dependencies and exchange relations among them. This raises the possibility of constructing explicit design information exchange relations, thereby enabling traceability, automation, and visualization of design processes that were previously unmanageable.

This paper presents an initial investigation into this possibility. Specifically, we aim to organize a conceptual methodology for building design information exchange relations using LLMs and to illustrate the approach with concrete scenarios and examples drawn from ship engine room design. The purpose is not to provide a fully realized system, but to demonstrate the feasibility and outline the technical challenges that must be addressed to advance toward practical implementation.

2. Related Work and Positioning of This Study

Ship design information has long been managed at the level of drawings and specifications. While drawings efficiently convey a holistic product view, they treat design information at coarse granularity. Dependencies among detailed elements remain implicit, making it difficult to trace change impacts, manage schedules, or reuse knowledge.

Research has therefore explored fine-grained management of design information. *Sato et al. (2023)* applied statistical correlation analysis (Cramer's V) to ship specifications, organizing results in a Design Structure Matrix (DSM) and clustering related elements. This clarified potential dependencies and modular groupings, supporting cross-departmental coordination, *Sato (2023). Cooper et al. (2011)* captured the U.S. Navy ship design process in detail using a DSM-based tool, representing task dependencies as matrices and networks. The study showed that fine-grained models can externalize tacit workflows, though knowledge capture required significant expert effort, *Cooper et al. (2011).*

Parallel efforts aim to extract and organize design knowledge. *Dong et al. (2022)* built a process knowledge graph for welding and assembly, mapping CAD features onto an ontology to externalize procedural know-how. This allowed case-based reasoning for reuse, though ontology development was labor-intensive. *Hiekata et al. (2010)* applied ontology-based text mining to fabrication reports, extracting "component–trouble" pairs and normalizing terminology. This revealed recurring design issues for feedback, but depended on expert-crafted ontologies. *Feng (2008)* proposed an ontology-driven multi-agent framework for collaborative ship design, showing potential for unified knowledge sharing but raising issues of scalability and maintenance.

Recently, large language models (LLMs) have been explored for design knowledge extraction. *Han et al. (2025)* combined LLMs with graph databases in GraphRAG, building a maritime knowledge graph with improved accuracy and efficiency over LLM-only approaches. *Doris et al. (2024)* introduced DesignQA, a benchmark for multimodal LLMs on specifications and CAD drawings, finding even advanced models like GPT-4V struggle with complex technical interpretation. *Jiang et al. (2025)* showed that retrieval-augmented LLMs using patents improved feasibility of design outputs but constrained creativity, highlighting a trade-off between accuracy and novelty.

Prior research demonstrates the value of fine-grained management (DSM, MBSE) and knowledge extraction (ontologies, graphs), yet practical application in shipbuilding remains difficult. Methods require heavy expert knowledge capture, standardization, and are hard to scale to vast, heterogeneous data. Meanwhile, LLM-based approaches show promise in automating extraction and relation inference from unstructured sources. This study positions itself as a preliminary step toward leveraging LLMs' contextual understanding and reasoning to automatically extract design information units and construct design information exchange relations, aiming to reduce dependence on manual knowledge capture and enable fine-grained, automated design management.

3. Definition of Design Information Exchange Relations

In this study, design information exchange relations are defined as the explicit links that describe which design information is referenced and which design information is influenced by a given piece of information. In other words, these relations capture the directional dependencies among design elements across documents, drawings, and specifications.

In ship design, each drawing both relies on information from other drawings and supplies information that will be used elsewhere. Clarifying these two directions of dependency makes it possible to see what must be prepared before a drawing can be created and what will later be required by others. Once the necessary information has been identified and is available, the drawing task can proceed even if other parts of the source drawings are not yet complete.

The overall flow of constructing such relations follows three principal steps:

1) Target drawings or design documents are selected as the scope of analysis.
2) Design information units are extracted from each drawing, such as parts, dimensions, specifications, materials, or equipment data.
3) Exchange relations among these extracted units are organized so that references and dependencies are made explicit.

On the basis of these identified relations, design management can be carried out in a systematic manner. For example, design schedules and progress can be managed by referring to the prerequisite and dependent relations among information items, thereby ensuring that downstream tasks begin as soon as their required inputs are available.

Ultimately, the framework may extend beyond drawings themselves to include knowledge embedded in design manuals, specifications, and standards that are referenced during the design process. By doing so, design information exchange relations can serve as a comprehensive map of the information flows that sustain ship design, thereby enabling greater traceability, modularity, and manageability of design activities.

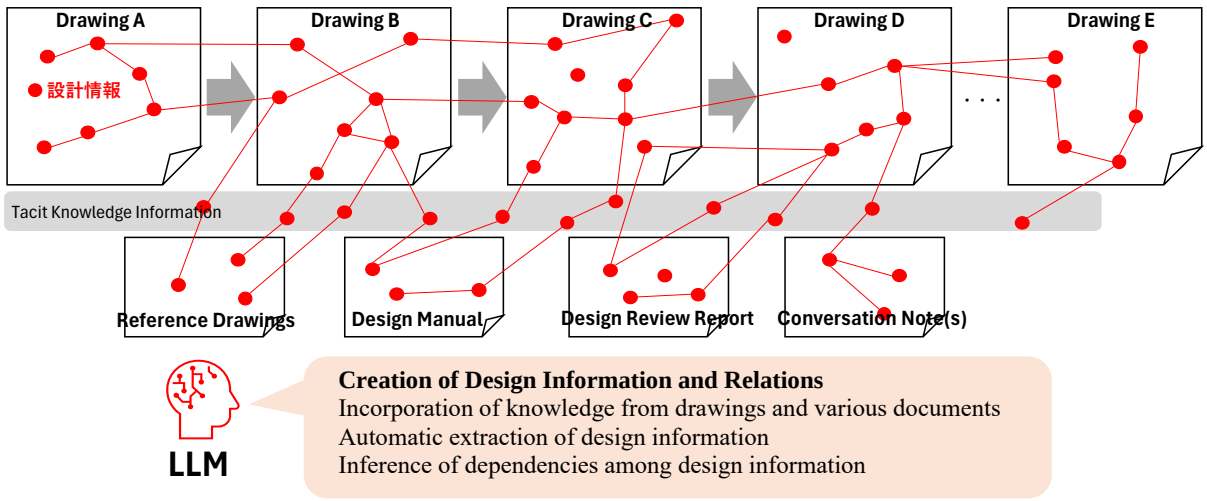


Fig.1: Conceptual diagram of design process based on design information units

4. LLM-based Framework for Constructing Design Information Exchange Relations

4.1. General Framework

The construction of design information exchange relations using large language models (LLMs) can be conceptualized as a multi-stage process that begins with the reading of design materials and culminates in the explicit mapping of dependencies among information units. The process is not confined to textual information alone but must also account for the graphical content of engineering drawings, which often conveys essential geometric and relational information.

In the first stage, design materials such as drawings, specifications, and manuals are ingested. Information embedded in these sources includes both textual annotations and graphical elements such as shapes, dimensions, and arrangements. Through a combination of optical character recognition,

shape recognition, and contextual interpretation, these elements are decomposed into finer-grained units of information. Each unit is then represented as a chunk and vectorized into an embedding space, enabling semantic similarity search and clustering.

In the second stage, the extracted information units are organized into design categories such as machinery, piping, structure, and electrical systems. At this stage, LLMs play a critical role in classification: they normalize varied terminology, resolve ambiguities in abbreviated or domain-specific expressions, and assign each chunk to an appropriate category by interpreting its context.

In the third stage, exchange relations among information units are inferred. Here, LLMs are used to evaluate whether two units are related, and if so, in which direction the dependency flows. For instance, an equipment specification may determine a pipe dimension, while a structural detail may be constrained by a piping arrangement. LLMs can be prompted to recognize such relationships, guided by domain rules and examples, and to output the most plausible relation type and direction.

Taken together, this framework highlights the distinct roles of LLMs across different stages. In category classification, LLMs are primarily employed as classifiers and normalizers, assigning chunks to structured categories based on contextual understanding. In relation inference, LLMs serve as reasoning agents, assessing the directionality and type of dependency between information units. By integrating these capabilities, design information exchange relations can be systematically constructed from heterogeneous, multi-modal design sources.

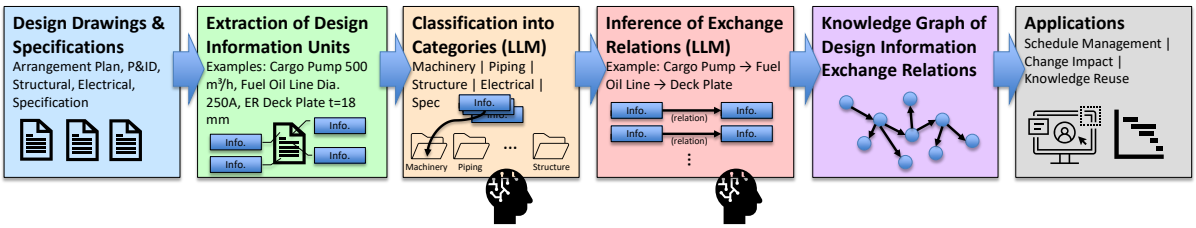


Fig.2: LLM-based Construction of Design Information Exchange Relations

4.2. Illustrative Demonstration with a Simplified Scenario

To provide a concrete image of the framework outlined above, we present a simplified, proof-of-concept demonstration. The purpose of this example is not to propose a complete methodology, but to give readers an intuitive understanding of how design information units can be extracted, categorized, and linked. For this purpose, we consider the case of a medium-size chemical tanker and focus on its engine room design. Several representative drawings are used as the target materials, including the engine room arrangement plan, piping and instrumentation diagrams (P&IDs), structural drawings of the engine room deck and bulkheads, and electrical routing plans. In this demonstration, the extraction of design information is carried out manually for clarity, though in practice this step would involve automated parsing and recognition tools.

From these drawings, a set of design information units is identified. Table I shows examples of extracted information, covering machinery, piping, structure, and specifications. These units include equipment capacities, pipe dimensions and materials, structural thicknesses, and operating conditions.

Once the information units have been extracted, they are categorized using LLMs. In this step, the LLM resolves terminology variations (for example, abbreviations such as “FO Line” or “SWC Pipe”) and assigns each chunk to a consistent category such as machinery, piping, structure, electrical, or specification. The next step is to infer design information exchange relations. Here, the LLM is prompted to judge whether two units are related, and if so, in what direction. The result is a network of dependencies that captures how one piece of design information constrains or enables another. Table II shows examples of such relations derived from the extracted units.

Table I: Examples of extracted design information units

ID	Design Information	Source Drawing	Category (by LLM)
M1	Cargo Pump Capacity 500 m³/h	Engine Room Arrangement	Machinery
M2	Ballast Pump Capacity 800 m³/h	Engine Room Arrangement	Machinery
M3	Main Engine Output 9480 kW	Engine Room Arrangement	Machinery
M4	Diesel Generator Capacity 1200 kVA	Engine Room Arrangement	Machinery
P1	Fuel Oil Line, Dia. 250A, SUS316	FO System P&ID	Piping
P2	Fuel Oil Line Pressure Rating: 1.0 MPa	FO System P&ID	Piping
P3	Sea Water Cooling Pipe, Dia. 300A, SUS304	SW Cooling P&ID	Piping
P4	Ballast Line, Dia. 250A, SS400	Ballast System P&ID	Piping
S1	ER Deck Plate, Thickness 18 mm, AH36	ER Structural Drawing	Structure
S2	Bulkhead Plate, Thickness 20 mm, EH36	ER Structural Drawing	Structure
S3	Foundation for Diesel Generator, SS400	ER Structural Drawing	Structure
E1	Power Cable 450 V to Diesel Generator No.1	Electrical Arrangement	Electrical
E2	Control Panel for Cargo Pumps	Electrical Arrangement	Electrical
SP1	Operating Pressure of Fuel Oil System: 0.8 MPa	Design Specification	Spec
SP2	Design Temperature of Sea Water Cooling: 32 °C	Design Specification	Spec

Table II: Examples of inferred relations among design information units

From (Source)	To (Target)	Relation (by LLM)
Cargo Pump Capacity 500 m³/h (M1)	Fuel Oil Line, Dia. 250A (P1)	Pump flow determines pipe diameter
Ballast Pump Capacity 800 m³/h (M2)	Ballast Line, Dia. 250A (P4)	Pump capacity determines ballast line size
Main Engine Output 9480 kW (M3)	Foundation for Diesel Generator (S3)	Main engine load requires reinforced foundations nearby
Main Engine Output 9480 kW (M3)	Girder G-12 (Structure, not in Table 1)	Engine power implies hull longitudinal stiffness
Diesel Generator Capacity 1200 kVA (M4)	Power Cable 450 V to Diesel Generator (E1)	Generator output specifies cable capacity
Diesel Generator Capacity 1200 kVA (M4)	Foundation for Diesel Generator (S3)	Generator weight/output defines foundation strength
Fuel Oil Line, Dia. 250A (P1)	ER Deck Plate, Thickness 18 mm (S1)	Pipe size requires reinforced deck penetration
Fuel Oil Line Pressure Rating: 1.0 MPa (P2)	Operating Pressure of FO System: 0.8 MPa (SP1)	System pressure defines pipe pressure rating
Sea Water Cooling Pipe, Dia. 300A (P3)	Bulkhead Plate, Thickness 20 mm (S2)	Pipe penetration requires stiffened bulkhead
Ballast Line, Dia. 250A (P4)	Transverse Beam T-5 (Structure)	Pipe routing influences beam arrangement
Control Panel for Cargo Pumps (E2)	Cargo Pump Capacity 500 m³/h (M1)	Control logic tied to pump operating capacity
Power Cable 450 V to Diesel Generator (E1)	Cable Tray Width 300 mm (Electrical)	Cable size influences tray width
Operating Pressure of FO System: 0.8 MPa (SP1)	Fuel Oil Line Pressure Rating: 1.0 MPa (P2)	Pressure condition constrains pipe design

From (Source)	To (Target)	Relation (by LLM)
Design Temperature of SW Cooling: 32 °C (SP2)	Sea Water Cooling Pipe, Dia. 300A (P3)	Temperature condition constrains material
Vibration Limit for Main Engine Foundation: 5 mm (Spec)	Foundation for Diesel Generator (S3)	Vibration criteria define foundation design
Corrosion Allowance for Deck Plate: 2 mm (Spec)	ER Deck Plate, Thickness 18 mm (S1)	Corrosion margin influences plate thickness
Ballast System Flow Rate Requirement: 800 m³/h (Spec)	Ballast Pump Capacity 800 m³/h (M2)	Flow requirement defines pump specification
Ballast System Flow Rate Requirement: 800 m³/h	Ballast Line, Dia. 250A (P4)	Flow requirement defines pipe diameter
Air Pipe to Starting Valve of Main Engine	Main Engine Output 9480 kW (M3)	Starting air supply dimensioned by engine requirements
Steam Pipe to Heating Coil	Boiler Capacity 6 t/h	Boiler steam capacity determines heating coil demand

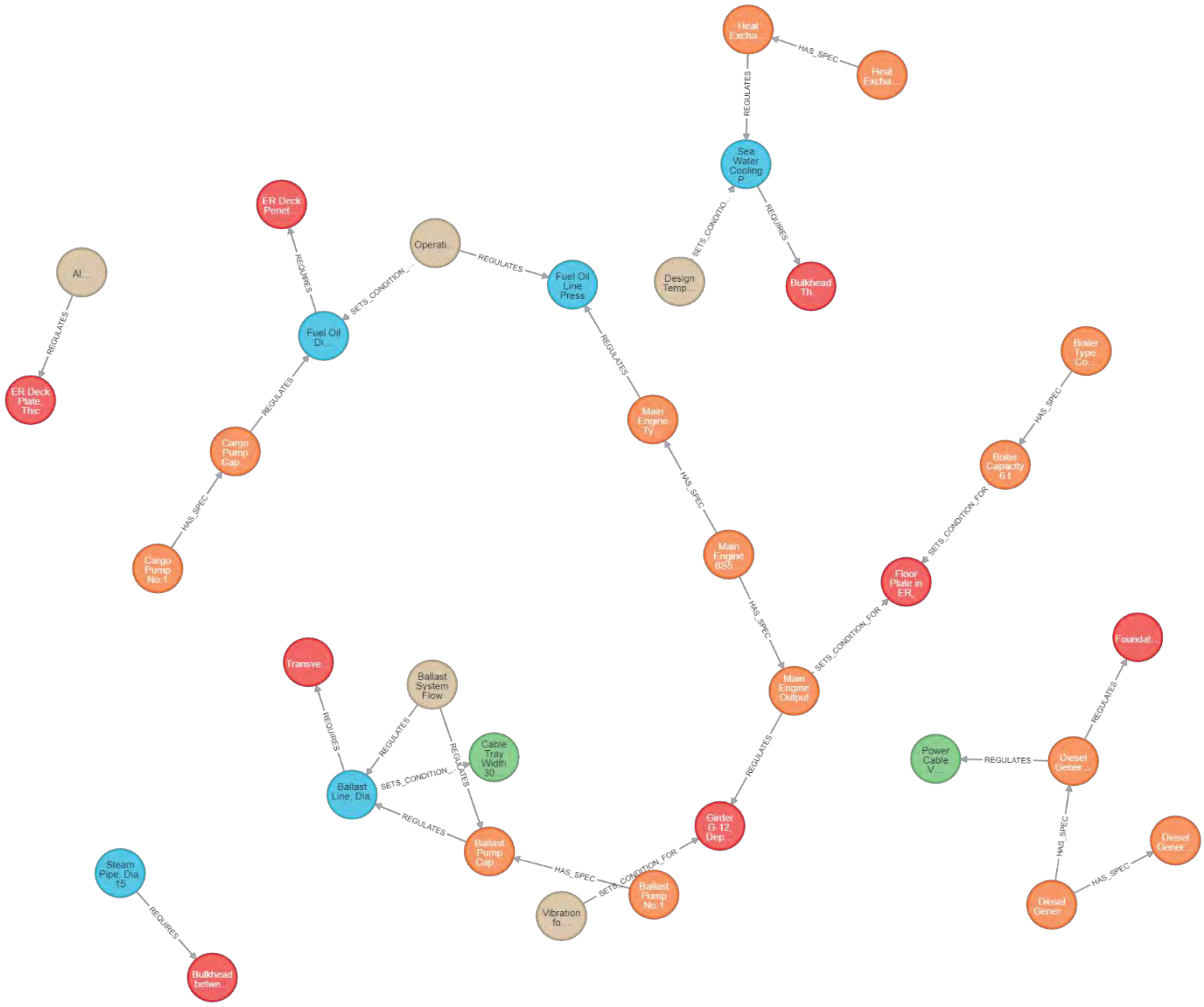


Fig.3: Image of a graph database of design information exchange relationships

Finally, the extracted information units and their exchange relations are stored in a graph database, where nodes represent design information and edges represent exchange relations. Fig.3 illustrates the resulting knowledge graph. The graph reveals chains of dependencies such as Cargo Pump → Fuel

Oil Line → Deck Plate, or Diesel Generator → Power Cable, enabling designers to visualize how design changes propagate across disciplines. Such a representation highlights the potential of LLM-supported methods to integrate heterogeneous design data into a coherent, navigable network.

5. Technical Challenges

The demonstration in Section 4 illustrates the potential of LLMs in constructing design information exchange relations, but it also reveals key technical challenges that need to be solved for practical application. These challenges can be grouped into two main areas: (i) extraction of design elements from drawings, and (ii) inference of exchange relations.

(i) Extraction of design elements from drawings
- Graphical primacy of drawings: In ship design, drawings communicate meaning mainly through shapes, configurations, and spatial arrangements, while text serves as supplementary annotations. Extracting design information requires linking textual labels with the corresponding graphical elements.
- Volume of design elements: Drawings may contain a vast number of possible elements. It is unrealistic to extract all of them indiscriminately; the extraction scope should be controlled according to the purpose of analysis.
- Granularity of information units: The level of chunking is critical. For example, a pump could be treated as one design unit, or divided into attributes such as capacity, material, and installation frame. The appropriate granularity depends on what kinds of exchange relations are targeted.
- Transition to 3D CAD models: With the growing use of three-dimensional CAD, richer but more heterogeneous data are available. Extracting meaningful design elements from 3D geometry, attributes, and metadata requires new strategies and integration of multimodal analysis.

(ii) Inference of design information exchange relations
- Multiplicity of definitions: “Exchange relation” can mean different things—reference between drawings, causal dependency (e.g., flow rate determines pipe size), or workflow sequence. The definition must be clarified for each application.
- Explosion of relations: If relations are traced too finely, the resulting network may become saturated, where nearly all elements are connected. Thresholds and criteria are needed to determine which relations are significant for design management.
- Variability in practice: Dependencies may change with design processes or the individual designer’s approach. While some general principles can be standardized, others remain context-specific and resist full formalization.
- Balancing generalization and specificity: Practical systems must combine standardized relation templates with flexibility to accommodate project-specific or organization-specific practices.

In short, two fronts of progress are necessary:
- Robust extraction of design elements from complex, multimodal sources (2D drawings, 3D CAD, specifications).
- Reliable inference of relations that are precise, meaningful, and manageable in scale.

Addressing these challenges will be crucial for transforming the concept of LLM-based design information exchange relations into a practical foundation for design management in shipbuilding and other complex engineering domains.

6. Conclusion

This paper has discussed a conceptual framework for constructing design information exchange relations in ship design by leveraging large language models (LLMs). We began by identifying the limitations of the traditional drawing-centric paradigm, where design information is managed at a coarse level and dependencies among information units remain implicit. We then defined design information exchange relations as explicit links that capture which information is referenced and

which is influenced, thereby making visible the prerequisites and impacts across drawings and documents.

Building on this definition, we outlined a general framework in which LLMs are employed at multiple stages: to assist in the extraction and normalization of design information units, to classify them into appropriate design categories, and to infer directional exchange relations among them. To illustrate the concept, we presented a simplified scenario based on an engine room design for a chemical tanker, demonstrating how information can be extracted, categorized, and connected into a graph of dependencies.

At the same time, we highlighted key technical challenges that must be addressed to move toward practical deployment. These include the extraction of design elements from graphical as well as textual content in drawings, the control of information granularity and scope, and the reliable inference of exchange relations that are both meaningful and tractable in scale. Balancing generalization with the variability of design practices remains a central issue.

The findings of this study suggest that LLM-supported methods have the potential to transform design management in shipbuilding by enabling traceability, visualization, and automation of information flows that are currently opaque. Future work will focus on implementing prototype systems that integrate multimodal extraction from 2D and 3D design sources, refining relation inference strategies, and validating the approach against real-world ship design projects.

References

COOPER, S.; ALLEN, G.; SMITH, R.; BILLINGSLEY, D.; HELGERSON, D. (2011), *Ship design process modeling: capturing a highly complex process*, 13th Int. DSM Conf., Cambridge, Massachusetts, pp.14-15

DONG, J.; JING, X.; LU, X.; LIU, J.; LI, H.; CAO, X.; DU, C.; LI, J.; LI, L. (2022), *Process knowledge graph modeling techniques and application methods for ship heterogeneous models*, Scientific Reports 12, 2911

DORIS, A.C.; GRANDI, D.; TOMICH, R.; ALAM, M.F.; ATAEI, M.; CHEONG, H.; AHMED, F. (2024), *DesignQA: a multimodal benchmark for evaluating large language models' understanding of engineering documentation*, arXiv:2404.07917

FENG, X. (2008), *Ship collaborative design based on multi-agent and ontology*, 5th Int. Conf. Cooperative Design, Visualization and Engineering (CDVE 2008), LNCS 5220, pp.249-252

HAN, Y.; YANG, T.; YUAN, M.; HU, P.; LI, C. (2025), *Construction of a maritime knowledge graph using GraphRAG for entity and relationship extraction from maritime documents*, Journal of Computer and Communications 13, pp.68-93, doi:10.4236/jcc.2025.132006

HIEKATA, K.; YAMATO, H.; TSUJIMOTO, S. (2010), *Ontology-based knowledge extraction for shipyard fabrication workshop reports*, Expert Systems with Applications 37/11, pp.7380-7386

JIANG, Z.; LIU, A.; ZHANG, D.; XU, X.; DAI, Y. (2025), *Customization and personalization of large language models for engineering design*, CIRP Annals 74/1, pp.191-195

SATO, Y.; AOYAMA, K.; OZAWA, H.; TAKAHASHI, K.; MAEDA, S.; HIRAI, N. (2023), *Organization of ship design information using Cramer's V and genetic algorithm*, 25th Int. DSM Conf., Gothenburg, pp.68-76

The Extension on the Usage of OCX Beyond 3D Structural Approval

Myeong-Jo Son, NAPA, Helsinki/Finland, myeong-jo.son@napa.fi
Ludmila Seppälä, NAPA, Helsinki/Finland, ludmila.seppala@napa.fi
Francesco Oneto, NAPA, Camogli /Italy, francesco.oneto@napa.fi
Bae Jun Kwon, DNV, Oslo/Norway, Bae.Jun.Kwon@dnv.com

Abstract

Significant progress has been made in the development of technologies and processes for 3D model-based structural approval of ships using the Open Classification 3D Exchange (OCX), driven by cross-industry collaboration. Although OCX was originally designed for structural approval, there is growing interest in exploring how the format can be further utilized. This research investigates the extended applications of OCX beyond 3D structural approval. Case studies include engineering assessments such as Finite Element Modeling (FEM), as well as information and data sharing through integration with Product Lifecycle Management (PLM) systems and detailed design processes. Furthermore, we examine the use of OCX compartments to support the automatic derivation of compartment volumes and their integration with NAPA stability calculations, enabling a streamlined and continuous workflow for hydrostatic and damage stability evaluations.

1. Introduction

Since its introduction to the shipbuilding industry with a primary focus on 3D approval, *Astrup et al. (2022)*, the OCX Consortium has taken the lead in improving the reliability and interoperability of the OCX by establishing the OCX Interoperability Forum, http://www.3docx.org. In parallel, numerous joint development projects involving classification societies, ship designers, shipyards, and CAD vendors have explored the application of OCX in 3D approval. These collaborations have helped identify process changes, tool requirements, and practical workflows for adopting 3D methods, while also clarifying the situations where traditional documents and drawings remain more effective, *Luli et al. (2025)*.

2. Extension on the Usage of OCX

2.1. Literature Review

The OCX format was originally developed as a standardized method for exchanging 3D ship structural models for classification approval, with the primary goal of replacing traditional 2D drawings and reducing manual verification tasks in structural plan approval. As model-based approval workflows have matured, there is growing interest in extending OCX to support cross-disciplinary aspects of ship design. Previous studies have highlighted several promising applications, including CAD-to-CAD model transfer, *Gušani et al. (2023), Zerbst (2023),* FEM model generation, *Son et al. (2022), Puurula et al. (2024*), verification of International Convention on Load Lines requirements, *Astrup et al. (2023)*, and the comment exchange between the classification society and design tool *Luli et al. (2025)*.

In particular, *Astrup et al. (2023*) suggested extending the OCX schema using the W3C XML xs:import construct. In this paper, a simpler approach to achieve a similar goal is presented in the following.

2.2. Research Approach

This research explores extending the use cases of OCX beyond 3D structural approval of ships. We propose two complementary approaches that maintain full compatibility with the existing OCX schema. The first leverages the ocx:CustomProperties feature to add flexibility for specialized cases

(Approach 1), while the second focuses on extracting ship-design–relevant information directly from OCX files without the need to reconstruct geometries or structural models (Approach 2).

Introduced in OCX schema version 3.0.0 and supported in subsequent versions, ocx:CustomProperties addresses the limitations of rigid schema designs, where each value has a fixed place. Such rigidity complicates schema management, particularly for less common or optional values. By adopting a flexible, generic key-value approach for each identified element (ocx:IdBase), this mechanism allows users to preserve original identifiers while adding additional values to support rare or specialized business scenarios.

Fig.1 illustrates how ocx:CustomProperties is defined in OCX schema v3.1.0 (left) and demonstrates its application to structural objects (ocx:Plate and ocx:Stiffener) within an ocx:Panel (right). In the example, the same key—OCX_CORRSION—is assigned to both ocx:Plate and ocx:Stiffener, where the corresponding values represent corrosion deduction thickness. Multiple custom properties can be associated with a single identified element; for instance, the ocx:Plate also includes a custom property indicating tightness. The main limitation of ocx:CustomProperties lies in the absence of built-in semantic definitions, meaning that the interpretation of keys and values must be communicated separately. One possible way to address this limitation is to introduce an additional ocx:CustomProperty at a higher-level element—specifically, the ocx:Vessel defined at the beginning of the OCX file—where the key could be "readme" and the value would provide guidelines describing all key–value sets used in the file as shown in Fig.2.

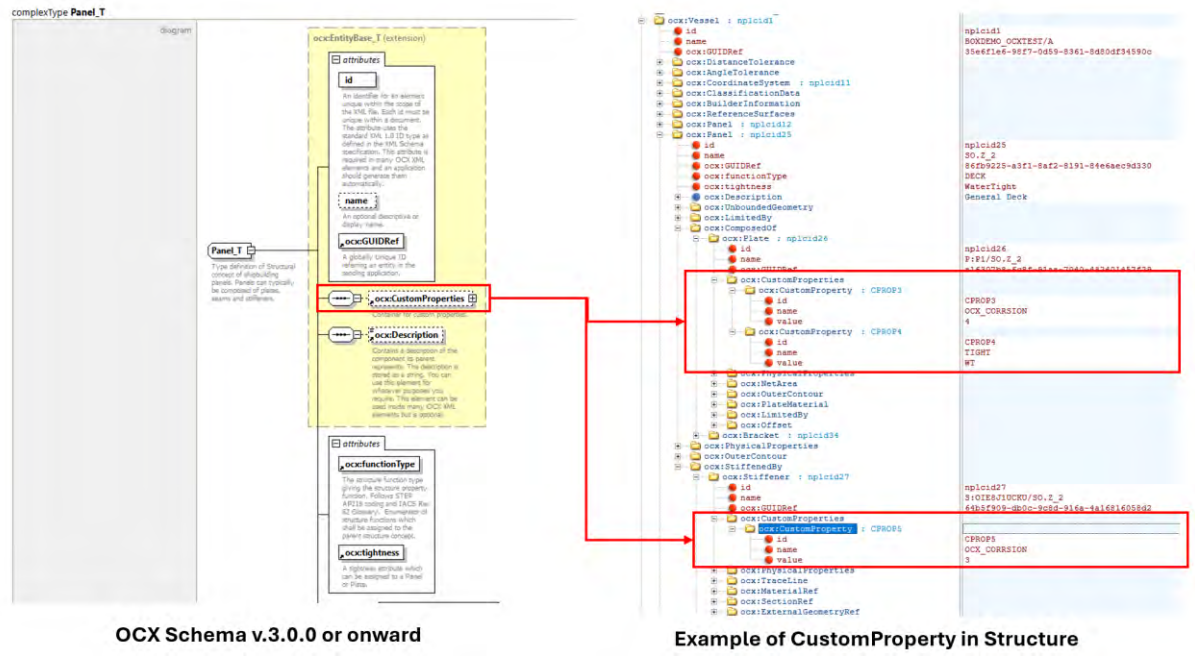


Fig.1: ocx:CustomProperty in the OCX schema with an example usage

Fig.2: Example of a self-explanatory ocx:CustomProperty defined at the ocx:Vessel level.

Another approach to extending the use of OCX is to extract valuable structured metadata directly from the OCX file. Since OCX already contains detailed information about ship structures, and also OCX is the structured XML (Extensible Markup Language) file, this data can be repurposed for various analyses without reconstructing the full geometry.

Fig.3 presents an example of how NAPA Designer can extract relevant information directly from an OCX file prior to reconstructing the structural model. In this case, the extracted data includes the OCX schema version, all function types, compartment purposes, as well as counts of panels, plates, openings/holes, stiffeners, brackets, pillars, compartments, and equipment that was presented by *Son et al. (2022).*

Examples can further include deriving weight distribution, plate area distribution for paint calculations, and lengths of stiffeners and seams for estimating welding requirements.

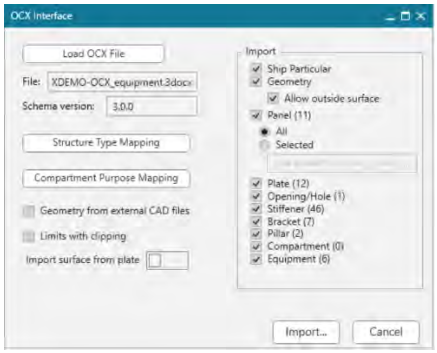


Fig.3: Example of extracting metadata without importing the OCX file

3. Case Studies in Structural Design

3.1. FEM

Since a FEM model is an idealized representation of a structural model, the current OCX file can also serve as a single source for FEM data when applying Approach 1. In shipbuilding, several FEM-interpretable file formats are commonly used depending on the finite element analysis (FEA) solver, including the Nastran Bulk Data File (BDF), ANSYS CDB file, DNV Sesam Input Interface File (FEM), Abaqus input file (INP). Although these formats differ in syntax, they share similar concepts: defining shell and line elements as node connections in absolute coordinates, along with their structural properties such as material, stiffener profiles, and plate thicknesses, *Son et al. (2018).*

In addition to the FEM model itself, it is essential to define groups of structural elements that can be further utilized for analyses such as loading conditions or corrosion reduction. These group definitions can be included directly within FEM-oriented formats (e.g., CDB or FEM files) or provided as separate files linked to the primary solver input of BDF file, such as a Patran session file (SES.01).

Since all relevant and accurate information for structural objects—and their hierarchy within panels—is already available in OCX, a FEM model can be incorporated into the OCX file with relative ease. This can be achieved by extending ocx:Point to include node identifiers and by adding elements defined as lists of node numbers with their IDs, linked to the corresponding structural objects such as plates, stiffeners, or brackets.

- Node - ocx:Point consists of two attributes: the coordinates in the form (x y z) and the unit. Since a FEM model typically uses a consistent unit system for all nodes, the FEM unit can be defined once at the beginning of the file. Each node can then be specified by its coordinates (x y z) and identifier, for example: node 1 – (ID1 x1 y1 z1). To further reduce complexity, nodes may also be expressed as a single custom property containing a list, e.g., (node 1, node 2, …, node n). This

custom property, defined as "FE_NodeList", can be placed at the ocx:Vessel level, as nodes represent global entities that are shared across the ship model as shown in Fig.4.

Fig.4: Node definition as ocx:CustomProperty in ocx:Vessel

- Shell Element - Shell elements can be classified into two types—quadrilateral or triangular—depending on the number of nodes. Since this can be determined when parsing the element definition, each element may be expressed in the form; element 1: (ID1 Node1 Node2 Node3 (Node4)). To simplify representation, multiple elements can also be combined into a single custom property as a list, e.g., (element1, element2, …, element n), defined as "FE_ElementList" as shown in Fig.5. This property can be assigned to an ocx:Plate, with the corresponding material information referenced directly from the existing ocx:PlateMaterial. When a stiffener is idealized as shell element, the property can likewise be defined in ocx:Stiffener or ocx:EdgeReinforcement. In such cases, the appropriate thickness must be determined by evaluating the element's normal vector to distinguish whether it represents the web or the flange of the stiffener.

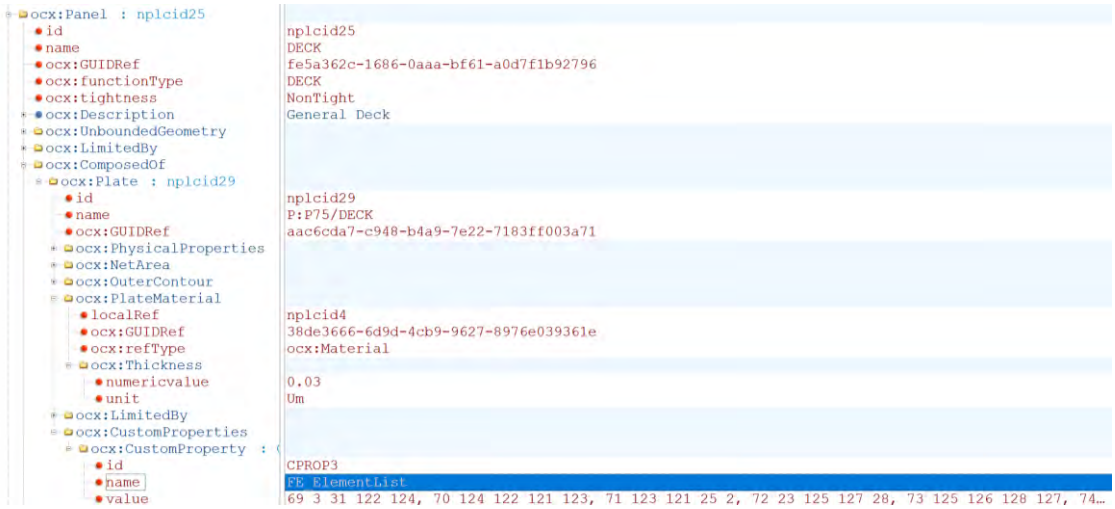


Fig.5: Shell element definition as ocx:CustomProperty in ocx:Plate

- Line Element - A line element consists of two nodes and can therefore be defined in a manner similar to a shell element, for example: Line 1: (ID1 Node1 Node2). Multiple line elements can also be grouped as a list under "FE_LineElementList" as shown in Fig.6. While the definition itself is straightforward, existing FEM input formats differ significantly in how stiffener profiles are represented across FEA solvers. By utilizing the standardized stiffener profiles already supported in OCX, these profiles can be interpreted consistently when OCX is adopted as an input format for FEM solutions. It is also common to idealize a stiffener by modeling the web as a shell element and the flange as a line element, resulting in both shell and line elements being associated with the same stiffener as shown in Fig.6. In this case, the flange's section properties should be derived directly from the original stiffener's section definition.

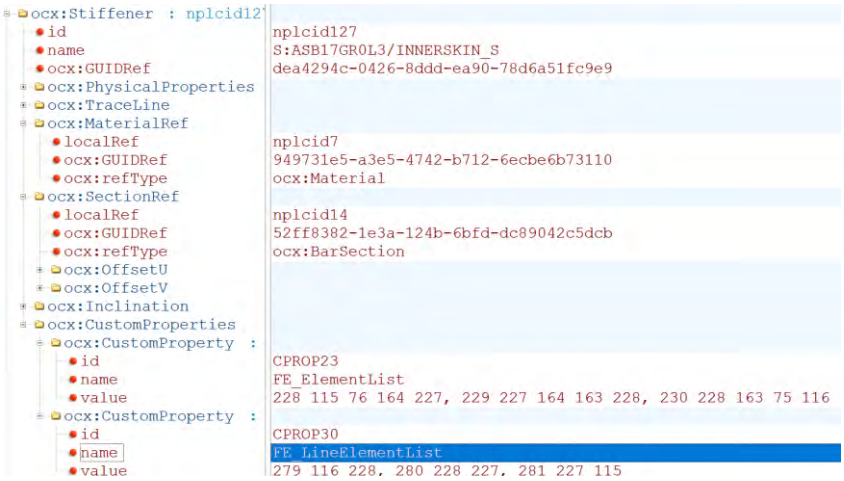


Fig.6: Line element definition as ocx:CustomProperty in ocx:Stiffener

The benefit of the proposed approach is that FEM groups can be naturally organized within the existing structural hierarchy, since each element is already associated with its parent steel object. Beyond these structural groups, additional FEM groupings by compartment provide significant added value for applications such as load definition and corrosion reduction. These compartment-based groups can also be included in OCX, as compartment information is already available within the same file.

A FEM compartment group can be defined as a list of element IDs, Fig.7. Two types of FEM compartment groups can be distinguished based on element location: elements located on the boundary of a compartment are classified as external, while elements located inside the compartment volume are classified as internal.

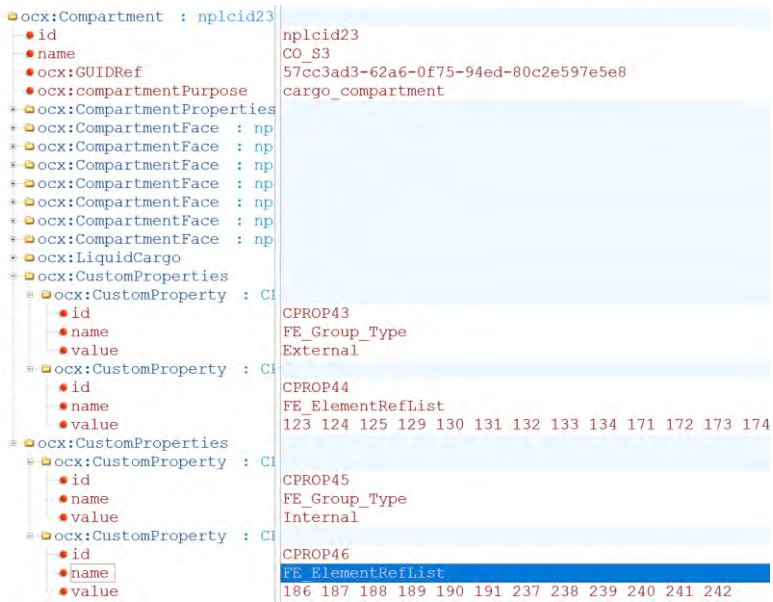


Fig.7: FEM compartment group definition as ocx:CustomProperty in ocx:Compartment

As a demonstration of the proposed approach, we developed a proof of concept that enables importing FEM models on top of the existing OCX import functionality in NAPA Designer. A single OCX file can contain structural data, compartments, and a FEM model defined as a custom property. When reading the file (left of Fig.8), the system first detects whether a FEM model is included. The user can then choose either to import the FEM model together with all structures and compartments into a new project, or to import only the FEM model into an existing project. A coarse-mesh FEM model, where stiffeners are idealized as shell elements, retains all relevant steel properties and can be grouped

according to their parent steel objects (middle of Fig.8). In contrast, a fine-mesh FEM model from a separate OCX file can be imported directly into an existing project (right of Fig.8).

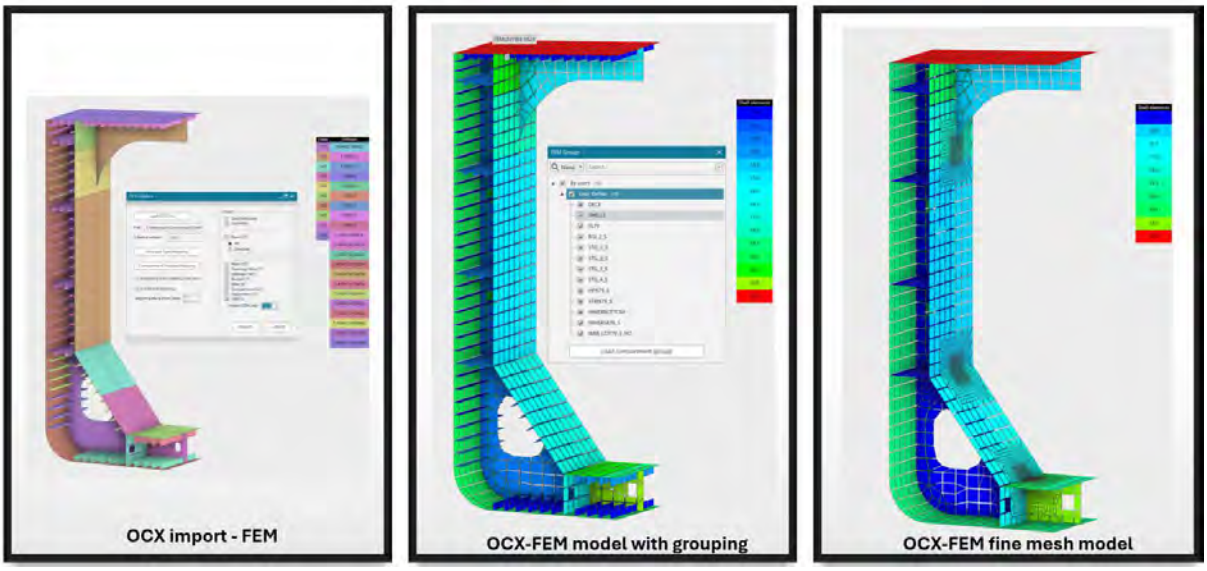


Fig.8: Demonstration of OCX FEM model interface

3.2. Information and data sharing

Using data for sharing between various applications and databases to facilitate the digital thread is another possible field of application for the OCX schema. As shipbuilding involves a large number of diverse stakeholders and a long life cycle of the engineering, design, construction, and operation phases, it is inevitable to encounter a variety of specialized and tailor-made applications. Relying on a standardized data structure can significantly improve information sharing and reduce the need for extensive integration interfaces. The most commonly used data ontology in shipbuilding is the SFI Coding and Classification System for Ship Information codification system, developed in the 1970s, *Manchinu and McConnell (1977)*. It provides a naming convention for eight main groups and a structured sub-group for all parts of the vessels, as well as a logic to link these codes with material purchasing and maintenance operations. While it is often criticized and typically used in a customized manner, the SFI is utilized for position IDs of equipment and outfitting, and it can frequently aid in structuring data for procurement and cost estimations.

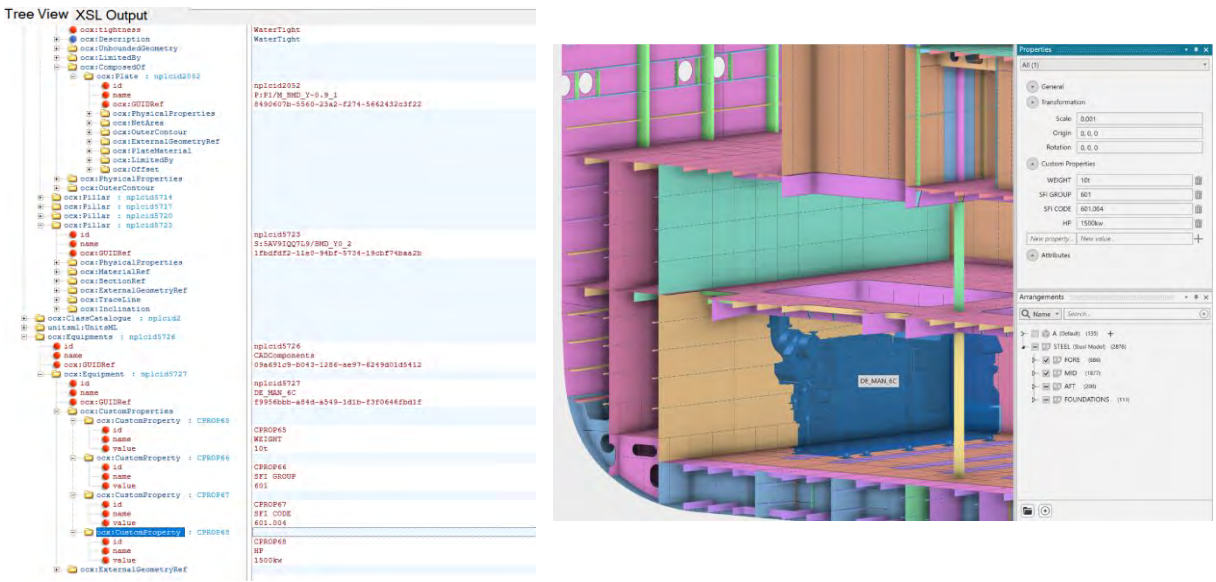


Fig.9: Example additional attributes to create SFI structure for 3D model components

Fig.9 shows equipment components represented in the 3D model, which were assigned custom properties of the SFI group and the SFI code. In the exported OCX file, these properties enable the recognition of the individual components, and the structuring of data as needed. This way, the OCX schema can evolve into a more comprehensive data taxonomy standard tailored to the shipbuilding industry, similar to BIM in the construction industry. For PLM initiatives in the industry, this categorization can open up possibilities for easier integration of authoring tools with data management and overall streamlining of data flows in product data management and in lifecycle data management. This structure can serve as a basis for identifying business objects between the data systems.

3.3. Steel output

3.3.1. Weight inclusion

An essential aspect of structural design is the estimation and the control at different stages of steel structure weight and center of gravity. In practical shipbuilding, not only the numbers extracted from a 3D structural model or measured from 2D drawings are sufficient.

The comparison of those values with shipyard statistic data is an essential doublecheck to prevent macroscopic errors or discover the weight trend for a prototype project. Since the usage of the 3D structural model is quite a recent resource, it is important that the weight list is presented in the same way it appears in the reference's ship documentation, maybe an old one or just following shipyard customs.

Usually, weight list functions allow filtering options based on: geometry constraints, structural type attributes and previously defined structural groups therefore matching a specific weight breakdown structure may be extremely difficult at least for some items.

The evolution of the software for structural modeling allows the designer to achieve a higher level of detail at every next model, getting the required consistency in weight grouping ends up into an additional challenge.

In a practical example, deck beams can be modelled as T bar extruded profiles for a quick structure definition and lately re-defined in the same model as: web + flange elements changing "de facto" the nature of the elements: from T beams as deck attributes to T beams as independent structures having their own attributes.

The experience shown that a certain amount of work is needed to re-organize the weight output in the desired way using Excel spreadsheets or similar tools.

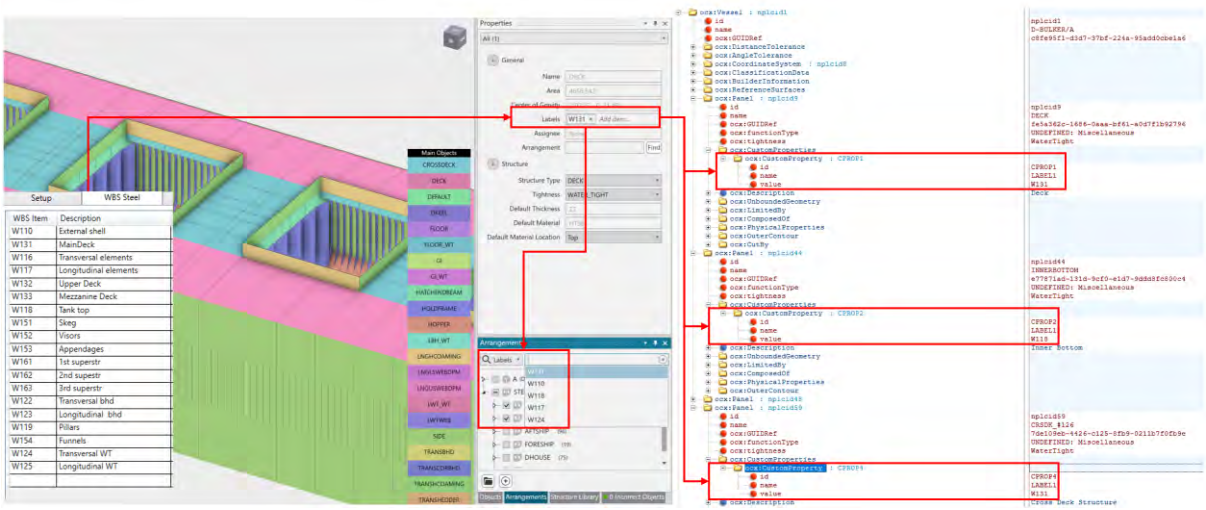


Fig.10: Label for customized weight inclusion as ocx:CustomProperty

The proposed solution is to use a totally free and customizable attribute of NAPA surface object called Label. An object can have one or more Labels without modifying the object's structural nature, Label assignment can be done in a very quick way using graphical interactive selection. After elements in a model are Labeled they can be grouped by Label as a filter. As NAPA Designer already exports the object with Label as ocx:Panel by assigning Label as ocx:CustomProperty, that any combination of element belonging to a weight group can be achieved and the result presented in a list does not require any post process as shown in Fig.10.

3.3.2. Material List

Another document that can be now achieved is the material list, it reports a breakdown of the steel structure divided by plates and profiles grouped by their significant attributes: plate thickness and material, profile scantling and material as shown in Fig.11. This provides a key example where Approach 2 is particularly effective.

Once more it not only represents an essential document to anticipate the material order, but also works as crosscheck measurement for the weight estimation consistency. It evidences the presence of possible errors in the model: unwanted profiles, duplicate profile types, incorrect material etc.

It is also possible to present the list result limited to each building block defined in the Production engineering schema.

Plate summary Data

Thickness (mm)	Material	Weight (t)	Area (m2)
7.5	AH36	25.418	431.732
13.0	AH36	51.404	503.711
9.0	AH36	134.849	1908.684
8.5	AH36	23.603	353.740
7.0	AH36	100.186	1823.227
8.0	AH36	164.437	2618.425
11.5	AH36	26.964	298.688
8.0	A	22.474	357.867
10.0	AH36	167.170	2129.553
12.0	AH36	0.889	9.442
12.5	AH36	2.017	20.560
7.0	A	97.778	1779.395
6.0	A	9.722	206.407
15.0	AH36	4.090	34.734
10.0	ALU	5.994	214.075
8.0	ALU	8.816	393.574
20.0	ALU	0.078	1.391
12.0	ALU	0.804	23.914
15.0	ALU	2.241	53.355
6.0	ALU	39.720	2364.275
Totals		888.654	15526.750

Profile Summary Data

Profile	Material	Weight (t)	Length (m)
HP240X10	AH36	7.501	294.583
HP140X8	AH36	28.141	2613.841
HP200X9	AH36	44.302	2389.340
FB150X15	AH36	10.747	608.491
HP160X9	AH36	3.499	250.790
FB120X10	A	4.093	434.458
HP120X6	AH36	22.972	3143.473
350X9/FL 150X12	AH36	5.355	137.802
300X9/FL 100X12	AH36	10.417	340.243
FB150X12	A	2.374	168.030
FB75X10	A	1.612	273.784
FB100X8	A	0.793	126.238
FB90X15	A	0.108	10.209
L120X60X8	ALU	11.508	2986.908
FB150X12	AH36	38.086	2695.373
FB100X8	AH36	1.131	180.146
HP140X8	A	6.153	571.493
FB100X12	A	1.477	156.763
HP120X6	A	15.051	2059.568
HP180X8	AH36	11.243	760.830
HP180X8	A	1.703	115.238
180X10/100X15	AH36	4.847	187.119
HP140X7	AH36	0.446	46.086
FB100X10	A	0.709	90.274
FB90X15	AH36	0.336	31.682
FB100X10	AH36	1.034	131.751
FB70X10	A	0.290	52.776
FB70X10	AH36	0.357	65.048
SHS160X160X8	AH36	4.769	124.899

Bracket Summary Data

Bracket	Material	Thickness (mm)	Weight (t)
TRT*700*700*0*180*50	AH36	8.0	0.258
	AH36	8.0	0.017
BD03A*S1*S1*300*125	A	8.0	1.292
BG02A*S2*S1*1000	A	8.0	0.399
BG02A*S1*S1*1000	A	8.0	0.395
TRT*350*350*0*50*50	AH36	10.0	0.055
TRT*350*300*0*50*50	AH36	9.0	0.005
BD03A*S1*S1*300*125	AH36	1.0	0.008
TRT*250*250*0*25*25	AH36	8.0	0.357
TRT*250*250*0*25*25	A	8.0	0.168
TRT*350*350*0*50*50	A	10.0	0.458
BG01A	AH36	9.0	3.857
BG2*S1*S2*10*130	AH36	8.0	0.465
TRT*350*300*0*50*50	A	10.0	0.256
Totals			7.991

Bracket Plate Summary Data

Thickness (mm)	Material	Weight (t)	Area (m2)
8.0	AH36	1.097	17.471
8.0	A	2.279	36.282
10.0	AH36	0.055	0.697
9.0	AH36	3.862	54.662
1.0	AH36	0.008	1.073
10.0	A	0.688	8.768
1.0	A	0.002	0.203
Totals		7.991	119.156

Fig.11: Example of material list grouped by thickness, profile, or bracket type

4. Extending OCX for Stability

Another area for the extension of OCX out of structural approval is stability analysis and its approval. Safety regulations, particularly damage stability, require detailed knowledge of watertight subdivision. At present, this information is often captured in separate 2D Watertight Integrity (WTI) plans and manually cross-checked against stability calculation models. This separation introduces inefficiencies and the risk of inconsistencies.

By extending OCX to explicitly represent watertight subdivision, the same 3D model can serve as the foundation for both structural approval and stability verification. This not only eliminates duplication of work but also enables earlier and more systematic compliance checking during design.

4.1. Watertight Integrity in Design and Approval

Watertight integrity refers to the arrangement of subdivision bulkheads, decks, doors, hatches, and other closures that prevent uncontrolled flooding after hull damage. Statutory regulations such as

SOLAS, MARPOL, the IBC Code, and classification society rules mandate specific requirements for watertight subdivision.

In practice, this means that ships must be divided into watertight compartments, and all openings in watertight boundaries (doors, manholes, ventilation ducts, etc.) must be fitted with closures. Failure to comply with these requirements can compromise the ship's damage survivability. A single overlooked non-watertight penetration could enable progressive flooding beyond the assumptions made in stability calculations.

Currently, WTI information is prepared as 2D plans. Class approval engineers check them manually, reviewing deck plans and cross-sections one by one. This process is time-consuming and susceptible to error, especially if late-stage design changes introduce new openings. The result is often a disconnect between WTI assumptions in the NAPA stability model and drawings used for approval.

The need for a digital transformation is clear: watertight integrity data must be embedded directly in the design model so that it can flow consistently into both class approval and stability analysis.

4.2 Proposed OCX-WTI Extension

4.2.1. Watertight Boundary

OCX already represents structural surfaces (e.g., bulkheads and decks) as ocx:Panel. In the OCX schema, ocx:Panel includes an optional attribute ocx:tightness, which can take the values NotTight, WaterTight, GasTight, or Undefined. This attribute can be defined directly in a structural CAD system such as NAPA Designer, where it is stored as a panel property, Fig.12. With this simple extension, any tool reading the OCX model can readily identify, for example, that Panel TBH is a watertight bulkhead panel.

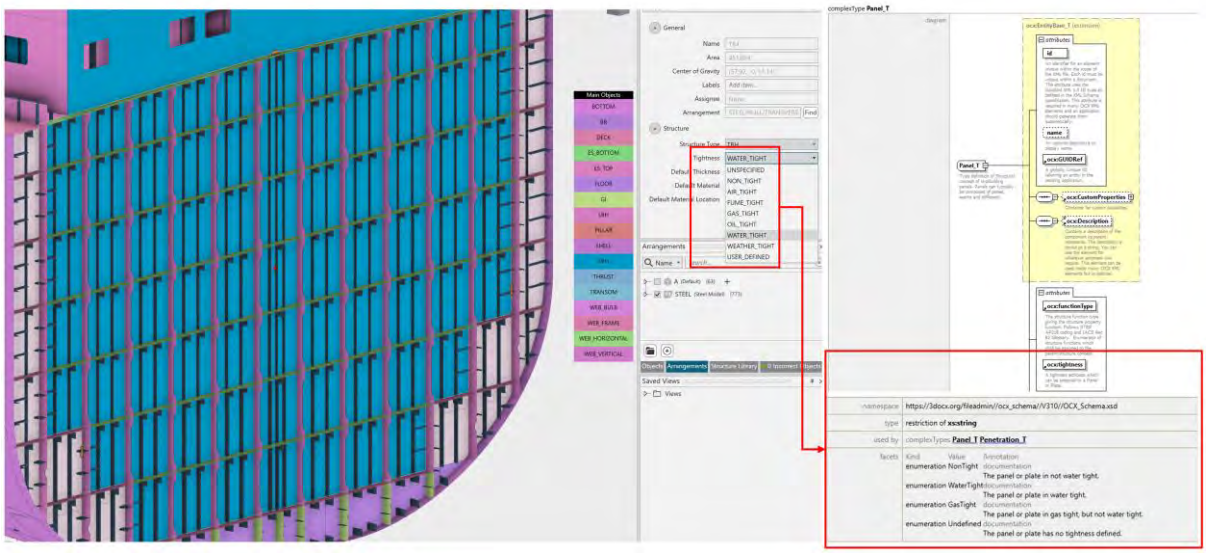


Fig.12: Tightness in the structural model and its representation as an attribute in ocx:Panel

4.2.2. Openings (Doors, Valves)

In ocx:Panel, the ocx:CutBy element is used to represent reductions and openings. Within this, components such as doors or valves can be defined in ocx:HoleContour. By extending these with ocx:CustomProperties, the type of opening (e.g., WatertightDoor, Valve) and its status (e.g., Closed, NormallyOpen) can be specified. Importantly, each opening is explicitly linked to the boundary it penetrates. This linkage enables automated consistency checks, for example, detecting and flagging an error if a non-watertight door is placed in a watertight bulkhead.

4.2.3. Compartment Connectivity

In the OCX schema, compartments are defined as closed volumes using ocx:Compartment. However, there is no explicit topological connection between a compartment and its corresponding ocx:Panel; instead, each compartment is described by a set of surfaces in the form of ocx:CompartmentFace. By introducing ocx:CustomProperties (e.g., "ocx:PanelRefID: deck"), linked to the relevant tightness attribute as described in Section 4.2.1, compartment connectivity can be validated. For instance, when two compartments share a bulkhead, the boundary can be explicitly defined as watertight (or non-watertight) based on the panel's tightness. Any missing tag would trigger a warning, indicating a potential unintended flooding path.

4.3. Practical Example

Consider a ferry with a car deck (Compartment C1) adjacent to an engine room (Compartment C2). In the OCX-WTI model as shown in Fig.13:

- C1 and C2 share a bulkhead.
- The bulkhead faces are tagged as watertight.
- A watertight door (Opening Door_A1) is located in this bulkhead with status = "Closed."

If a designer later changes Door_A1 to a weathertight door, the inconsistency is automatically flagged. Similarly, if the door is reported open during operation, the stability monitoring system can detect a violation of watertight integrity.

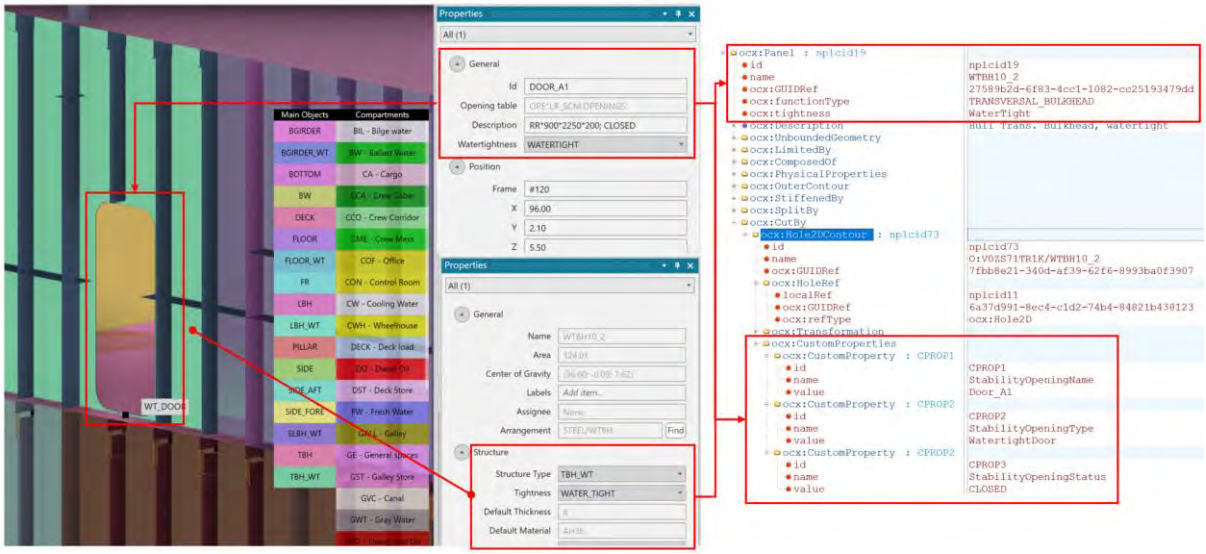


Fig.13: Alignment of structural and stability openings using ocx:CustomProperty

4.4. Integration with NAPA Stability

One of the most powerful benefits of OCX-WTI is direct integration with stability solvers such as NAPA Stability.

A NAPA script can import the OCX-WTI file, build the compartment network, assign permeability, and automatically generate damage cases. For example, "flood compartments 5 and 6 through a hull breach" can be simulated directly from the OCX definition.

This eliminates the traditional problem where the CAD model, the class approval model, and the NAPA stability model drift apart. With OCX-WTI, the same subdivision definition serves all purposes - structural approval, stability analysis, and onboard monitoring.

In operations, the as-built OCX-WTI file could be linked with real-time data (tank levels, door sensors). If a door required to be closed is detected as open, the onboard stability system can cross-verify against OCX-WTI and immediately alert operators.

4.5. Future potential

4.5.1. Rule-Based damage Case generation

OCX-WTI can also be applied to automate the deterministic damage stability process mandated by frameworks such as ICLL, MARPOL, and the IBC Code. In these regulations, the damage extent is prescribed according to principal ship dimensions, and designers traditionally have to interpret the rules manually when defining flooding cases. By embedding this logic directly into the OCX-WTI model, the workflow becomes far more efficient and consistent.

The process begins by encoding the rule-based damage extents, typically expressed in terms of ship length, breadth, and depth defined in OCX schema. Once these extents are formalized, the 3D OCX hull model allows for automated “sliding box” placement along the vessel, systematically identifying which compartments are affected by each prescribed damage. The watertight integrity data included in OCX-WTI ensures that all relevant bulkheads, decks, and closures are properly considered, and the resulting compartment flooding scenarios can be exported directly to solvers such as NAPA Stability.

This integrated approach eliminates the need for repetitive manual setup, guarantees compliance with deterministic stability rules, and ensures that case generation is both traceable and reproducible. Moreover, the digital model enables early-stage validation at the concept design phase, providing a single source of truth that links design, approval, and stability analysis. In this way, OCX-WTI elevates deterministic case generation from a semi-manual procedure to a structured, rule-compliant, and automated framework for stability assessments

4.5.2. Probabilistic Damage Stability and Monte Carlo Methods

In addition to deterministic MARPOL and IBC rule-based damage case generation using the OCX-WTI, the probabilistic damage stability framework in SOLAS requires ships to demonstrate survivability across a wide range of potential damages. The baseline is the zonal approach: the ship is divided by watertight bulkheads, combinations of damaged zones are generated, and each case is weighted by a probability (p-factor) and survival factor (s-factor) to calculate the attained index A.

Monte Carlo methods complement this by generating thousands of random damage scenarios based on statistical distributions of damage length, penetration, and vertical extent. Unlike zonal cases, these are sampled freely within the hull geometry, offering finer resolution of survivability and sensitivity studies.

OCX-WTI unifies both approaches by embedding watertight compartments, openings, and boundaries directly in the 3D model. Solvers such as NAPA can automatically extract subdivision data to generate zonal cases, ensuring full coverage and consistency with actual geometry. In Monte Carlo sampling, the same geometry is used so that flooding scenarios respect real connectivity and watertight boundaries, avoiding unrealistic damage paths.

Beyond bulkheads and doors, OCX-WTI can encode penetrations, valves, and closing appliances. This allows solvers to distinguish between permanent and operational boundaries, ensuring that only genuinely compromised conditions—such as an open valve—affect the analysis.

Through this integration, OCX-WTI transforms probabilistic stability assessment from a manual, error-prone process into a consistent and automated workflow. It secures compliance with SOLAS requirements while enabling advanced risk-based evaluations, elevating OCX-WTI from a simple tagging scheme into a comprehensive rule-compliance framework for stability.

This would transform OCX-WTI from a tagging scheme into a comprehensive rule-compliance framework for stability.

4.6. Operational Applications and Vulnerability Assessment

Beyond design and approval, OCX-WTI data can be extended into the operational phase. Once the as-built model is available, watertight integrity information embedded in OCX provides a digital baseline for real-time monitoring.

- Link to onboard sensors: Door status, valve operations, and tank levels can be continuously mapped against OCX-defined boundaries.

- Vulnerability assessment: Operators (e.g., cruise and ferry companies) can perform "what-if" flooding analyses in real time. For instance, if a watertight door remains open, the OCX-WTI–linked stability solver can instantly show the impact on survivability and evacuation routes.

- Decision support: This enables proactive risk management, supporting SOLAS Reg. II-1/13,13-1,17 and 17-1 requirements for limiting progressive flooding, and offering ship operators a clear picture of residual stability under both intact and damaged conditions.

- Digital twin in operation: The OCX-WTI model thus evolves into a safety digital twin, continuously updated with live data, forming the basis for vulnerability assessment and emergency response planning.

This operational extension aligns with the increasing demand from passenger ship operators for tools that enhance safety awareness, regulatory compliance, and decision-making during daily voyages and emergency scenarios.

5. Conclusion and future work

In this paper, we explored OCX from new perspectives to examine how it can provide additional benefits throughout the ship design and operation process. Using two alternative methods—(i) extending the schema with ocx:CustomProperty on relevant elements and (ii) extracting meaningful metadata without reconstructing the 3D model—we demonstrated applications such as FEM model transfer, preparation of input for PLM systems, and generation of steel lists for weight and material calculations.

Beyond structural aspects, we also investigated extensions of OCX that support 3D approval for ship stability by incorporating attributes such as tightness and openings. This approach illustrates how OCX can evolve from a structural approval data model into a multidisciplinary digital backbone for ship safety, enabling:

- Verification of vessel watertight integrity using 3D models supported by automation,
- Early identification of subdivision issues during concept design,
- Direct integration with NAPA and potential onboard digital twins, and
- Automated generation of deterministic and probabilistic damage cases.
- Extending OCX-WTI into vulnerability assessment for enhanced damage control and ERS support

The overall aim is to reduce manual work, avoid duplication in model exchange, and minimize human error by positioning OCX as the interface for a single source of truth. For shipyards and designers, this means earlier feedback and fewer late-stage design changes. For classification societies, it offers reduced review effort and improved confidence in model accuracy. For operators, it opens

opportunities for digital twins that directly connect design intent with real-time safety monitoring, thereby enhancing both safety and efficiency.

In addition, there is significant potential in extending the use of 3D structural data into the stability domain, offering several promising directions for future work:

- Schema integration with onboard stability computers (e.g., Type 4) to leverage OCX watertight connectivity for more accurate and reliable progressive flooding models.
- Standardized volumetric calculations, including gross and net tonnage assessments, enabling direct extraction from the digital model rather than relying on traditional manual methods.

References

ASTRUP, O.C.; AAE, O.; KUS, T.; UYANIK, O.; BITERLING, B.; BARS, T.; POLINI, M.; VIJAYA, G.; YU, K.; SEPPÄLÄ.T.; SON, M.J. (2022), *Moving towards model-based approval – the open class 3D exchange (OCX) standard*, Int. J. Maritime Eng. Part A.

ASTRUP, O.C.; AAE, O.; GRØNLIE, A.; UYANIK, O.; GIGERNES, S. (2023), *Enhancing the 3D Approval Process using Functional Zones in Ship Design*, COMPIT Conf., Drübeck

GUŠANI, S.; RADONIĆ, M.; PUURULA, J. (2023), *OCX Standard and Structural Model Reuse in the Shipbuilding Design*, COMPIT Conf., Drübeck

PUURULA, J.; PUDD, M.; GUŠANI, S. (2024), *Advantages of Model-Based FEA Reporting Integrated with OCX Approval*, ICCAS Conf., Genoa

LULI, C., SON, M.-J.; ASTRUP, O.C.; MADSEN, A. (2025), *3D Model-based Approval in Reality*, COMPIT Conf., Pontignano

MANCHINU, A.; McCONNELL, F. (1977), *The SFI Coding and Classification System for Ship Information*, REAPS Technical Symposium, Louisiana

SON, M.J.; LEE, J.H.; PARK, H.G.; LEE, J.Y. (2018), *Mobile Visualization for Finite Element Model and Assessment Result of Whole Ship*, COMPIT Conf., Pavone

SON, M.J.; SEPPÄLÄ, T.; MERIKANTO, J.; AAE, O.; ASTRUP, O.C. (2022), *Utilization of OCX as Part of 3D Model Based Approval in Ship Design Process*, COMPIT Conf., Pontignano

ZERBST, C. (2023), *OCX on the Way from Research to Industry Practice*, COMPIT Conf., Drübeck

A RAG-based LLM Approach for Data Validation and Harmonization in Ship Design

Janica Bronson, NTNU, Ålesund/Norway, Janica.a.bronson@ntnu.no
Maryam Teimouri, UTU, Turku/Finland, mtebad@utu.fi
Henrique Gaspar, NTNU, Ålesund/Norway, henrique.gaspar@ntnu.no
Icaro Fonseca, NTNU, Ålesund/Norway, icaro.fonseca@ntnu.no
Karolina Bierkowska, NTNU, Ålesund/Norway, karolina.bierkowska@ntnu.no
Filip Ginter, UTU, Turku/Finland, figint@utu.fi
Herbert Koelman, NHL Stenden, Leeuwarden/The Netherlands, herbert.koelman@nhlstenden.com

Abstract

Validating ship design data across systems is challenging due to fragmented information from multiple sources, file types, and formats – from 2D drawings, 3D models, and specifications, often found in unstructured text files. While unified 3D models aim to serve as a single source of truth, ensuring accuracy and consistency across all representations remains a complex task. This paper presents a retrieval-augmented generation (RAG) solution for extracting and comparing design parameters from diverse files and formats. The approach aims to detect inconsistencies between documents and versions, helping designers maintain data integrity and reduce manual effort throughout the ship design process.

1. Introduction and Cost of Errors in Early Ship Design

The early concept design stage in naval architecture represents a critical phase where extensive data generation occurs under severe time constraints. This stage is characterized by intensive multidisciplinary collaboration and competitive bidding processes that require simultaneous development of numerous design documents across specialized domains, *Andrews (2018)*. Despite the inherent uncertainty and reliance on preliminary estimations, concept designs must rapidly converge to meet stringent bid requirements and project timelines. This phase exhibits unique constraints: (1) compressed timeframes with intense pressure, highly collaborative workflows requiring specialized expertise, *Le Poole et al. (2023)*, continuous validation of design parameters against performance thresholds, *Brathaug et al. (2008)*, and substantial uncertainty in design assumptions and calculations, *Jorge et al. (2018).*

The convergence of these factors creates an environment highly susceptible to errors that can propagate through subsequent design phases, with correction costs escalating exponentially as detail levels increase, as shown in Fig.1, *DeNucci and Hopman (2021).* Research indicates that early-stage design errors can result in cost overruns when discovered during detailed design or construction phases, and sometimes irreparable errors leading to high repercussions, *Andrews (2021), Rigterink (2014).* This sensitivity necessitates robust validation mechanisms to ensure parameter consistency and accuracy throughout the iterative design process.

Current industry practice relies heavily on traditional design spiral methodologies and concept variation methods (CVM) that involve multiple manual review cycles and version synchronization processes, *Papanikolaou (2018)*. However, these approaches face fundamental limitations in modern ship design environments, characterized by the fragmented nature of design information that spans 2D drawings, 3D models, specifications, and unstructured text files, *Bronson et al. (2024)*. While unified 3D and collaborative environments are increasingly promoted as single sources of truth, ensuring accuracy and consistency across all design representations remains a complex challenge, *Koelman et al. (2024)*. Studies reveal that engineers spend approximately 14% of their time locating information and verifying accuracy, representing a significant inefficiency in time-critical design phases, *Chui et al. (2023)*. Existing version control and change tracking mechanisms prove inadequate for managing the rapid iteration cycles characteristic of concept design. Moreover, current validation approaches require extensive manual synchronization between different systems and file formats.

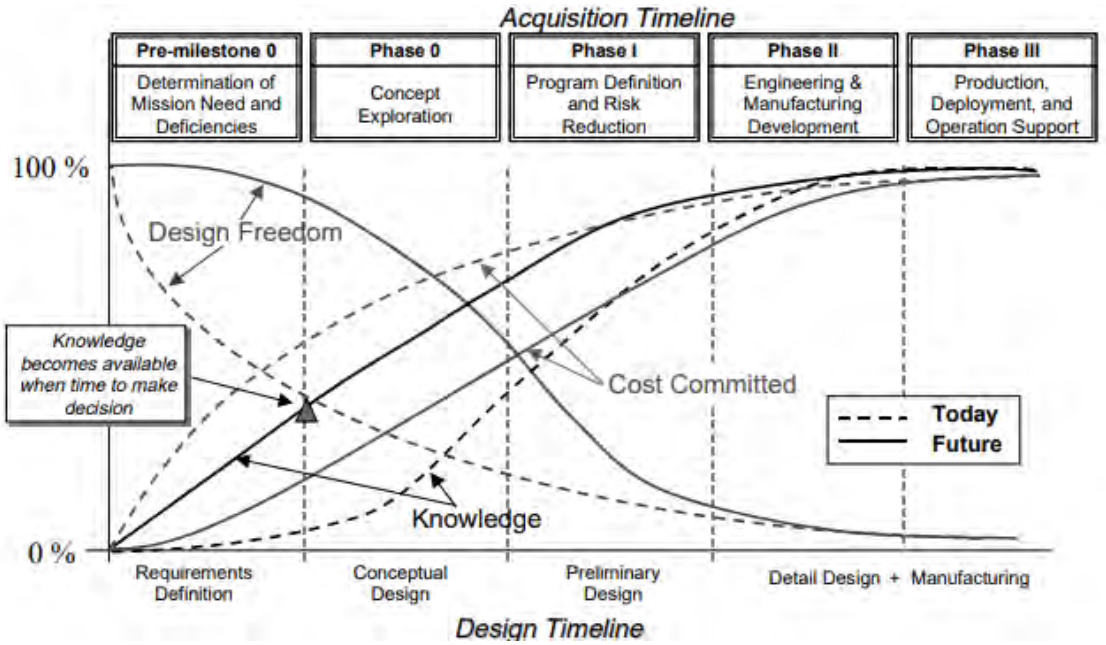


Fig.1: Relationship of committed costs and design freedom (adopted from *Mavris and DeLaurentis (2000)*)

This paper presents a novel approach involving retrieval-augmented generation (RAG) that addresses these fundamental challenges by automatically extracting and comparing design parameters from diverse file types and formats to reveal data inaccuracies and potential errors. Unlike traditional synchronization-based solutions that require comprehensive workflow restructuring and lengthy implementation periods, this method focuses on reviewing inconsistencies across documents and versions while preserving existing design processes. The key advantage of this approach is its non-disruptive integration with the current workflow; designers can continue using their preferred tools and established practices, while the solution is available to designers to validate parameter consistency.

2. Current Practice and Challenges

Despite the proliferation of advanced digital design environments, early-stage ship design validation remains heavily reliant on manual cross-referencing of heterogeneous data sources. Designers are often required to consult and compare design parameters from CAD models and hydrostatic calculations to spreadsheets and regulatory documents. This process is not only labor-intensive and error-prone but also constrained by tool interoperability and limited access to proprietary software platforms.

A core challenge lies in the fragmentation of design data across multiple formats and systems. Critical parameters - including principal dimensions, form coefficients, stability margins (e.g., GM), and design coefficient - are distributed across lines plans, general arrangements (GA), structural models, weight estimates, and machinery specifications. These parameters exhibit strong interdependencies: for example, hull form characteristics influence hydrostatic stability; structural arrangements affect weight distribution; propulsion requirements impact hull resistance and fuel consumption. Ensuring coherence across these dimensions necessitates continuous cross-document validation, which current workflows do not adequately support.

Moreover, regulatory compliance further complicates validation. Requirements elucidation is a core task that involves the synthesis of multiple regulations and guidelines from class. Design proposals must align with diverse and evolving standards, including SOLAS, MARPOL environmental regulations, Energy Efficiency Design Index (EEDI) thresholds, and classification society rules. These overlapping requirements generate a multi-objective validation landscape in which inconsistencies can propagate unnoticed, particularly when validation relies on manual inspection.

While modern software such as CADMATIC and AVEVA Marine supports model-based approvals, *AVEVA (2020), Yllikäinen (2019)*, they are primarily optimized for detailed design and approval stages - not early-stage concept design. Most tools validate geometry and compliance but overlook consistency across functional parameters. These and other validation techniques are discussed below:

1. Manual or in-house Validation Tools (Isolated) - Designers must manually extract and compare parameters from technical documents (e.g., line plans, hydrostatic reports, spreadsheets). This task is not only time-consuming but also highly susceptible to human error, especially as design iterations increase. Isolated scripts or digital checklists may help automate the validation of specific parameters (e.g., GM, LCG/LCB). While helpful, these solutions may not scale due to interoperability limitations.
2. Quality Assurance (QA) Procedures (Peer/External Review) - In many firms, validation is deferred to QA reviews. These reviews require cross-functional teams to manually synthesize inputs across disciplines, increasing the cognitive load. Although there is new research in this domain, there is a need for critical company buy-in for these QA processes and require dedicated personnel to carry through, *Hmeshah et al. (2015)*.
3. Model-Based Validation (OCX and Similar Standards) - OCX-based workflows and 3D model-centric platforms are designed to encapsulate validation within a unified geometric model. However, these models typically support only those parameters that can be directly visualized or geometrically mapped (e.g., structural members, arrangement boundaries). Alphanumeric parameters such as stability margins, performance coefficients, or other important design data currently still remains outside the scope of these models and must be validated separately, *Astrup (2022)*.
4. Software Tools – Class is also leading the development of new tools for validation. For example, the development of Nauticus Hull's Rules Check allows users to run their finite element analysis (FEA) against relevant cargo holding rules and thresholds, *DNV GL (2018)*. These tools, apart from DNV Nauticus Hull, include AMBS Eagle UDM, ClassNK PrimeShip-Hull, Lloyd's Register's RulesCalc, Korean Register's SeaTrust-HullScan, among others. However, these are mainly focused on structural validation.
5. Novel approaches – New solutions are being proposed by persons such as Soman (2015), who aim to improve the Smart Ship Design (S3D) environment by addressing the current lack of capability in evaluating design against engineering guidelines. The proposed solution uses Natural Language Processing (NLP) to extract design guidelines efficiently. However, the solution stops at the extraction level, *Soman et al. (2015)*, a gap this paper hopes to address.

3. Large Language Models (LLMs) and RAG

Large Language Models (LLMs), when combined with RAG, open new possibilities for assessing inconsistencies across technical documents. While LLMs provide context-aware reasoning over complex language, RAG enhances this capability by incorporating fresh, external data into the model's responses. By embedding and indexing technical documents, the system can instantly cross-reference them - allowing ship designers to ask questions such as, "Is the GM value consistent across all reports?" or "Does this hull design meet SOLAS and EEDI standards?"

Large pretrained language models are highly effective at retaining knowledge and retrieving factual information from their parameters. However, their effectiveness tends to decrease on downstream tasks that require expanding or updating their knowledge. Hybrid approaches that combine parametric memory with non-parametric memories can help address these limitations, as they allow knowledge to be revised and expanded more easily and quickly, *Lewis et al. (2020)*. *Siddharth and Luo (2024)* introduce a retrieval-augmented generation (RAG) framework specifically designed for ship design patents. It focuses on extracting named entities and their relational structures from patent texts to construct a structured, domain-specific knowledge base that supports more accurate and context-rich information retrieval.

Fig.2 illustrates the enhancement supported by RAG in the 3-step process of prompt or question answering, covering indexing of documents, retrieval of relevant documents based on semantic similarity, and input into the LLM for the generation of final answers. Existing research, such as *Soman (2015)*, is limited to filtering, highlighting, and extracting relevant rules from technical standards. By incorporating the generative reasoning of LLMs together with RAG's ability to retrieve and integrate external knowledge, these capabilities can be significantly extended. This enables ship designers not only to identify applicable rules but also to cross-validate them against new documents and evolving designs.

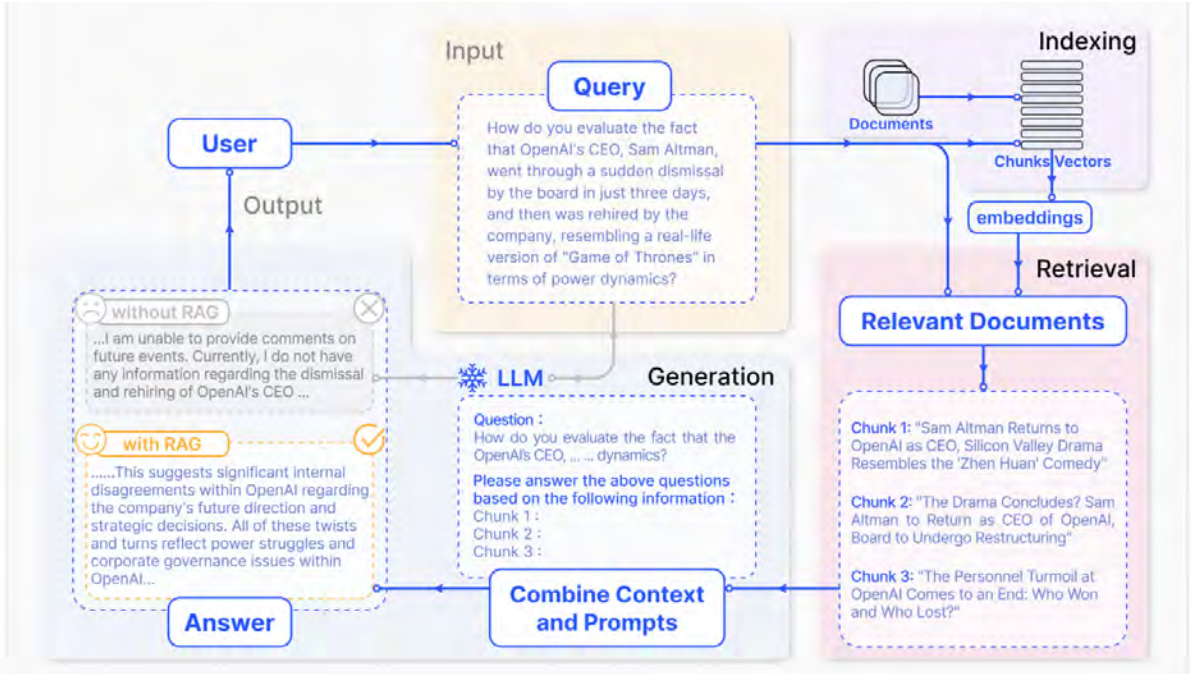


Fig.2: Representation of the RAG process enhancing LLM for prompting, *Gao et al. (2023)*

4. Case Study

To explore the potential of LLM-RAG validation approaches in early-stage ship design, we focus on a case study centered on the research vessel 'RV Gunnerus'. This case examines whether inconsistencies in design parameters can be effectively identified across multiple document types and subsequently reviewed against established design rules.

In this pipeline, we consider the usability of LLMs and RAG mainly in:

- Extracting parameters from unstructured PDFs and text files
- Cross-validating across internal design documents
- Highlighting inconsistent and regulatory requirements,

4.1. Parameters and Dataset

The study focuses on a core set of interrelated parameters commonly found in specification sheets such as principal dimensions (e.g., Length Overall, Beam, Draft), capacities, machinery data, equipment, and mission-specific facilities. Hence, the dataset used involved the 'RV Gunnerus' specification sheets, general arrangement (GA) drawings, hydrostatics inputs used for preliminary hydrodynamics tests, the 3D model, and equipment data. This original dataset contains: (1) the 3D model, (2) technical 2D drawings, and (3) text data in PDFs. The dataset was provided in part through research within the SEUS Project, which enabled access to NTNU ShipLab. For this case study, the data was used in its original form without any pre-processing.

In the next section, we discuss the key technical aspects of developing the AI assistant, including data preparation, system development, and model selection.

4.2. Methods

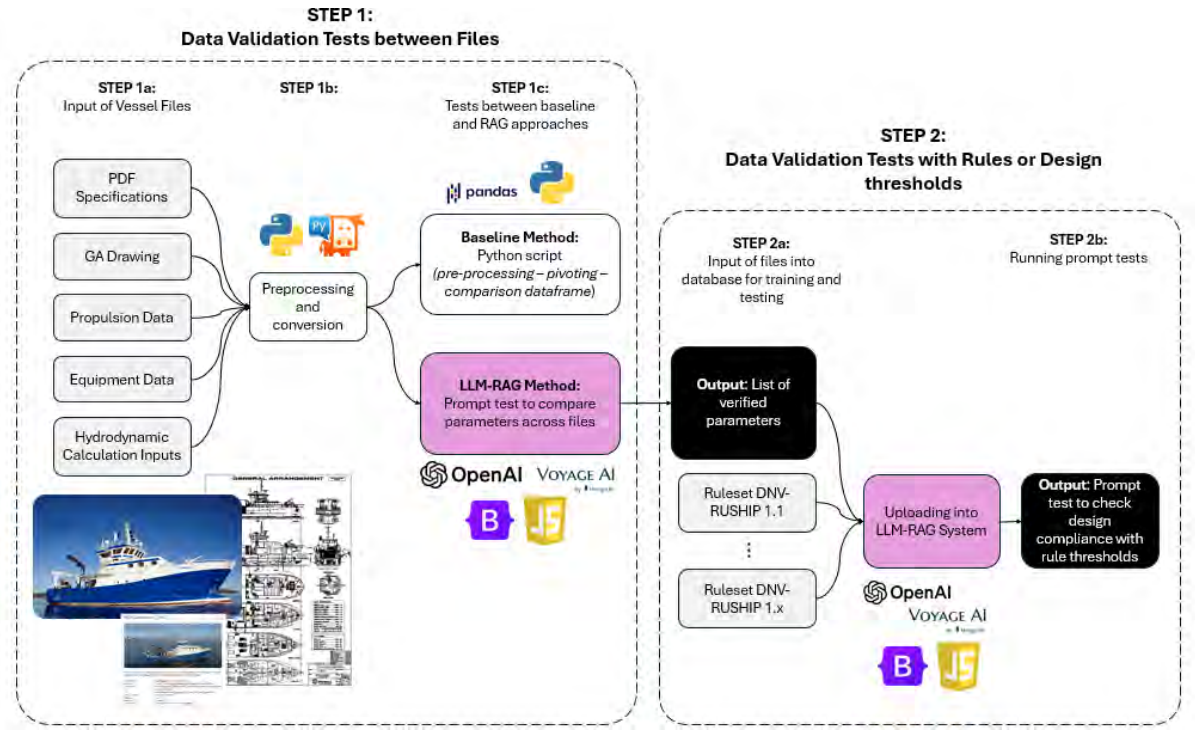


Fig.3: Methods for 2-step Validation of Parameters

The methods for this study employ a 2-step approach, as visualized in Fig.3, to cover the aforementioned goals:

1. Data Validation Tests between Files: For this step, the contents of the various design files are compared with each other. The files range from 2D, text, to 3D drawings and are pre-processed with the help of an LLM and further fed back into the model for prompt testing. In order to understand the gains in using the LLM-RAG model, we compare the results of this step with a Python script that automates the comparison of parameters between files.
2. Compliance and Validation of design parameters against rules: For this step, the vetted parameters are then compared to rulesets – an additional standard protocol for compliance and design validation. Given that *RV Gunnerus* has known notations, additional guidelines are fed into the model and used to assess whether the parameters are potentially compliant or not.

4.2.1. Data Processing

For data processing, the goal was to develop a scalable pipeline suitable for companies and organizations handling large volumes of data. Some of the data was already in text format, but a significant portion first had to be converted into images and then extracted as text, since the available metadata was not useful. To address this, we employed automated approaches using Python, with tools such as docx, pytesseract, PIL's Image module, and PyMuPDF (fitz).

As part of the data processing pipeline, we briefly investigated the integration of 3D model data into our system. The dataset provided contained primarily. prt and .x_t files, which are proprietary formats typically created with licensed software such as Siemens NX. These formats could not be processed directly using open-source Python libraries like pythonOCC. However, we identified converting these files into more accessible formats such as. STEP (.stp), STL, or. IGES would enable further processing

and analysis. Since Siemens NX supports scripting for large-scale batch conversions, this conversion step can be incorporated into an automated workflow without compromising the scalability of the pipeline. To fully integrate the 3D models into the RAG framework, we need to define the relevant keywords or structural features to extract from the 3D data to support meaningful retrieval and knowledge augmentation. Achieving this would also require developing additional scripts for calculating dimensional properties. However, to keep the scope of this paper focused and to avoid potential errors from miscalculations, we limit our work here to 2D PDFs and text data.

4.2.2. Data Comparisons

For comparing the data of the files, we developed a Python script to complement the RAG method and compare the effectiveness of the RAG system. The Python script collects the pre-processed data, appends them into a data frame, and pivots the data frame such that only unique parameters are indexed and the different sources are concatenated into columns. These values are then compared in order to determine if they are consistent or inconsistent. The output results reveal a summary report in text file format.

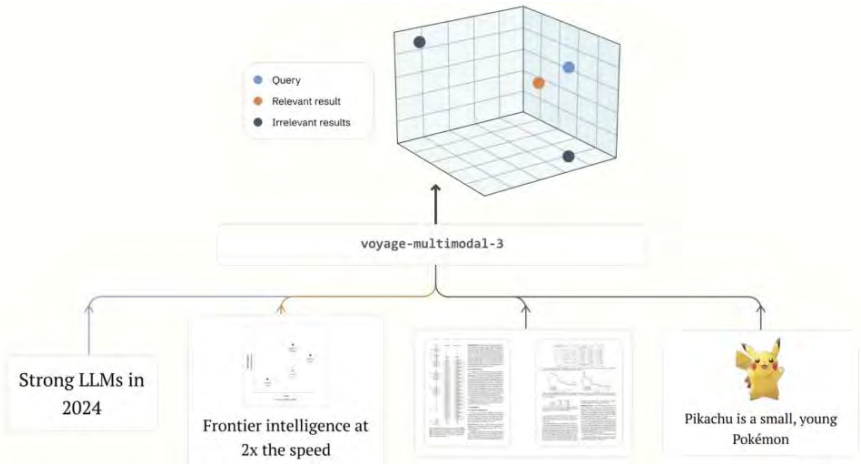


Fig.4: Latest Voyage AI Multimodal Framework, *VoyageAI (2024)*

4.2.3. RAG System

In designing the RAG system, we identified the need for two distinct machine learning models: one for embedding the data into a vector space for efficient storage and retrieval, and another as an LLM to generate coherent textual outputs. Given the multilingual nature of the dataset, it was essential to ensure that the system retrieves semantically relevant content based on context rather than language similarity. Through preliminary evaluations, the Voyage Multilingual model demonstrated the best performance in retrieving contextually accurate information across languages, *VoyageAI (2024)*. For text generation, we chose GPT-4o due to its state-of-the-art multilingual capabilities, low latency, and cost-effectiveness. These decisions build on the foundations laid by the original GPT-4 architecture, which demonstrated robust multilingual reasoning and generation across tasks, *Open AI et al. (2023).*

4.2.4. Web Interface

The web interface was developed using Python and FAISS for the backend, and Bootstrap and JavaScript for the frontend. The system is designed to provide a chat-based environment where domain experts can pose questions and receive responses generated from contextually relevant documents. Key components of the system are fully configurable, allowing users to select the number of top retrieved documents (n-top), the similarity metric, the embedding model, and the language model (LLM) used for generation. A document viewer is integrated into the right-hand panel, enabling users to validate the generated answers by reviewing the source documents. These documents can also be downloaded for further inspection.

Fig.5: Web Interface for the RAG Solution

4.3. Results

Tests for the first step reveal that about 173 unique parameters were found. In the Python baseline approach, about 168 were known consistent parameters and about 5 parameters were deemed inconsistent. These parameters include moulded breadth, cargo hold volume, fuel oil volume, tonnage, and depth data. The output summary file is shown in Fig.6. However, there are limitations to this method in that semantically equivalent parameters were not compared. For instance, 'loa' was not compared to length overall.

In comparison, the LLM model claimed to detect about 16 inaccuracies from the post-processed 173 unique parameters and was able to review semantic similarities, given the prompt: 'what inconsistencies do you find'. Fig.7 shows these parameters. It caught the same inconsistencies flagged by the Python script but also went further by identifying semantically similar terms and comparing them. For example, it correctly recognized that LOA and length overall refer to the same measure, as with LWL and length at waterline. It highlighted the differences in these values in the specification sheet and GA drawing. This discrepancy is likely due to the GA representing a different (lengthened) version of the vessel compared to the specification. The model was also able to distinguish between different types of water capacity, such as technical water versus potable water, and compare their values across documents - revealing, for instance, a 0.4 m^3 difference between the specification sheet and the GA.

However, the model also produced some less meaningful comparisons. In several cases, it compared terms against themselves (e.g., draught underside keel, net tonnage, trim, and water ballast volume), which resulted in misleading outputs. It also compared draught normative with draught at max load (conceptually different measures), and inconsistencies in the level of detail when describing the same propulsion equipment.

Overall, we find the RAG solution can determine more equivalent parameters, but we also observe that it tends to over-correct, showing high sensitivity to anomalies and often attempting to infer more differences than are actually present. Compared with the existing pipeline, we expect the RAG function to streamline the process by removing the need for manual execution of a Python script, offering greater convenience through the developed interface. The user-friendly interface also facilitates easier scrutiny

of results, unlike the Python script where comparisons are hardcoded and require users to review the code directly to ensure nothing was overlooked.

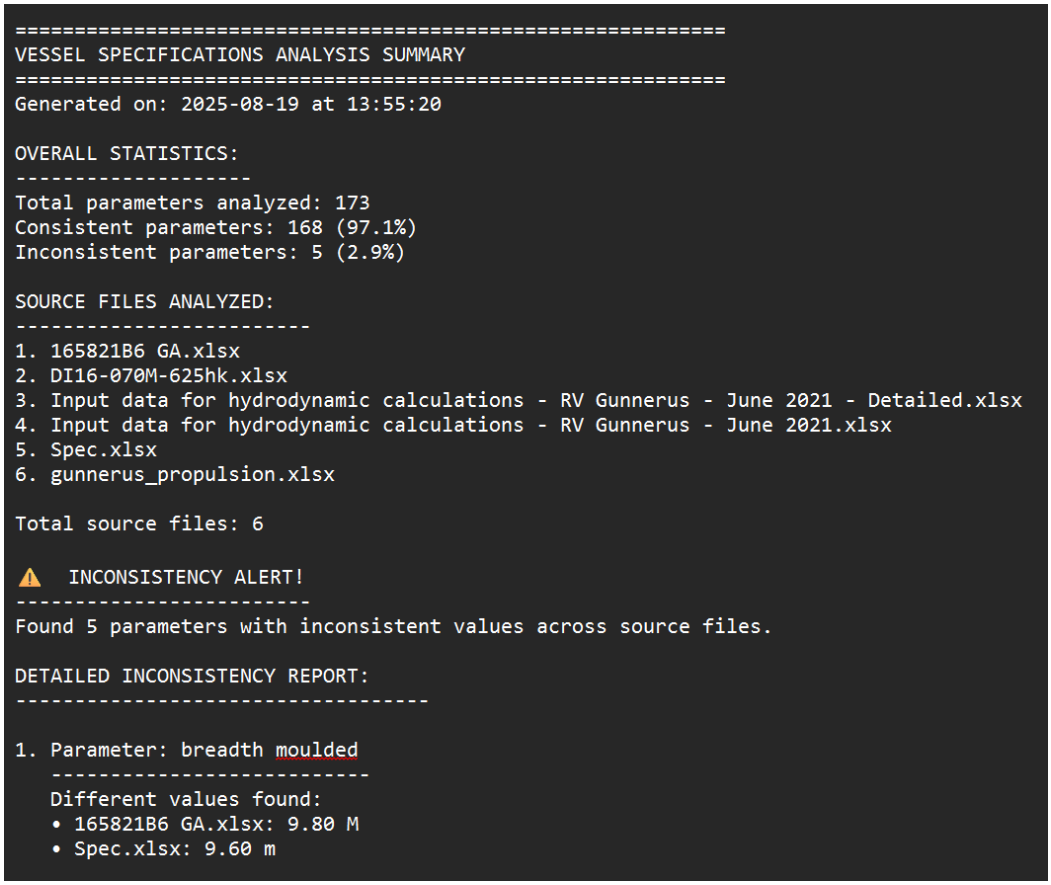


Fig.5: Results from Python Script

On top of the extraction piece, as per the tests for Step 2, we are also hoping to test how the RAG can be used to evaluate compliance of the given parameters against the rules. The vessel has the following rule notations: DNV + 1A1 + Ice C + E0 + R2 Cargo ship. The goal was to double-check how much the RAG can support design compliance, given the known and validated parameters against the rules. The model was populated with DNV rules DNVGL-RU-SHIP Parts 1 to 6 and the following prompt was ran: 'Does the RV Gunnerus specification comply with the attached DNV rules for a research vessel, given that it has the following rule notations <<DNV + 1A + Ice C + E0 + R2 Cargo ship>>?'

With the aid of breaking the prompt further down, the RAG-LLM was able to ascertain first what the notations means: 'The notation "1A + Ice C + E0 + R2" indicates that the cargo ship is classified with the following specifications: "1A" signifies a high ice class for vessels operating in ice-infested waters, "Ice C" denotes the ship's capability to navigate in light ice conditions, "E0" indicates the vessel has no restrictions on the use of electrical propulsion, and "R2" refers to the ship's compliance with specific requirements for reliability and redundancy in its systems.' However, for the specifications for each notation, the model advises consulting the latest DNV classification for the latest guidance, showing that while it can ascertain the rules, it is not able to sift through the parameters where thresholds were obvious. It is important to note that, at this stage, we do not expect the LLM to have the ability to perform more complex math and calculations in order to infer compliance.

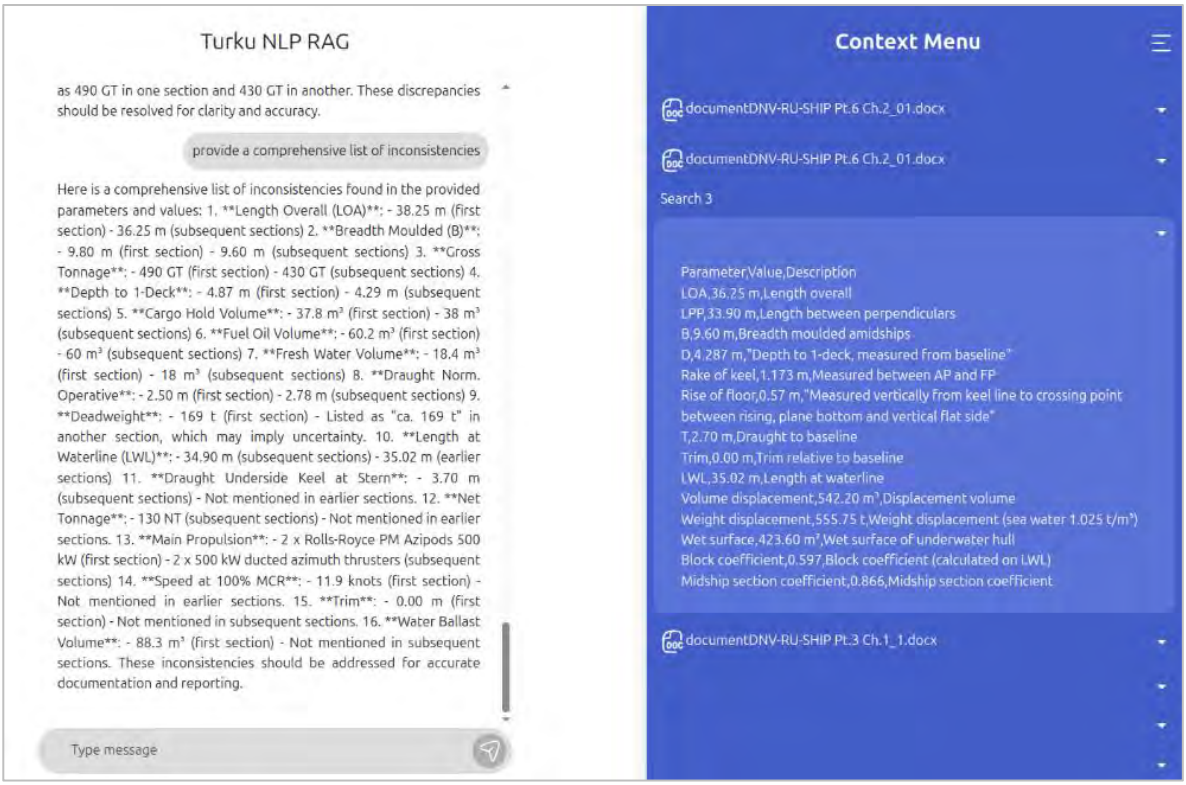


Fig.6: Results of STEP 1 and corresponding rulesets in the context sidebar

Fig.7 displays the interface starting with the prompts around design parameter inconsistencies, as introduced. The Context Menu demonstrates how these prompts are connected to the rule sets provided to the model. As additional rules and design data are incorporated, the interface has the potential to serve as a platform where designers can not only detect data inconsistencies but also evaluate them against the corresponding rules in the Context Menu – making the two validation steps not only possible but performed in parallel.

4.4. Discussions and next steps

There are various limitations and learnings from the case study that are subject to future improvements both for the model and the interface:

1. Metadata: Currently, the model only reads text and context data for retrieval. This makes it challenging to prompt the LLM to review more specific files and documents. The metadata of the files, including file name, version, and date of generation, is not currently considered, but would be helpful for future reference, allowing designers to easily point to specific documents.
2. Uploading interface: Alongside this development, a more user-friendly interface for uploading documents and retrieving them would be handy. To make testing and exploration easier, it would be helpful to add a drop box for uploading files that automatically updates the database.
3. Secure database for corporate data: Leveraging corporate data and deploying local instances of the model can tailor the LLM to a company's specific needs. By training it on corporate templates, designs, and terminology, the model becomes better aligned with organizational practices, making prompts for vessel types, project numbers, and other domain-specific inputs more intuitive and customized for the design team.

Noting these potential improvements, the addition of more and more data for training the LLM can increase the overall sensitivity and accuracy of the model. Further training is expected for the model so that it can become increasingly aware of maritime and ship design-specific semantics.

5. Conclusion

This paper has demonstrated the potential of an LLM-RAG solution for addressing one of the most persistent challenges in early-stage ship design: validating and harmonizing heterogeneous data. By comparing traditional scripted validation, we showed that the RAG system can capture not only explicit inconsistencies but also semantically equivalent parameters that are often overlooked.

While the current system shows sensitivity to anomalies and occasional over-correction, its ability to unify fragmented sources and provide designers with a transparent, user-friendly interface points to strong practical value. The approach does not disrupt existing workflows, reducing reliance on manual synchronization and enabling parallel checks for both parameter consistency and regulatory compliance.

Future work will focus on enriching the model with metadata, improving the upload interface, and tailoring models to organizational data for enhanced reliability. With continued refinement, the proposed solution can evolve into a scalable validation assistant, reducing design risks, lowering costs associated with late-stage errors, and ultimately accelerating the path toward more integrated digital ship design environments.

Acknowledgements

This paper is the result of the research project funded by the Smart European Shipbuilding (SEUS) Project (grant ID 101096224). SEUS is dedicated to advancing computational tools to enhance ship design and construction efficiency while also raising awareness of the current state of the art in maritime digitalization. Further insights and findings can be accessed at https://www.ntnu.edu/seus.

References

ANDREWS, D. (2018), *The Sophistication of Early Stage Design for Complex Vessels*, Int. J. Maritime Eng. 160(SE 18)

ANDREWS, D. (2021), *Design Errors in Ship Design*, J. Marine Science and Eng. 9(1)

ASTRUP, O.C. (2022), *Moving Towards Model Based Approval – The Open Class 3D Exchange (OCX) Standard*, Int. J. Maritime Eng.

AVEVA (2020), *AVEVA E3D Design*, AVEVA Inc.

BRATHAUG, T.; HOLAN, J.; ERIKSTAD, S. (2008), *Representing Design Knowledge in Configuration-Based Conceptual Ship Design*, 7th COMPIT Conf., Liege

BRONSON, J.A.; FONSECA, Í. ; & GASPAR, H.M. (2024), *Challenges Towards an Integrated Digital Twin Platform for Maritime Systems: Tackling Shifts in Data Ownership*, Offshore Technology Vol.1

CHUI, M.; ISSLER, M.; ROBERTS, R.; YEE, L. (2023), *Technology Trends Outlook*, McKinsey & Co

DeNUCCI, T.; HOPMAN, J. (2012), *Capturing Configuration Rationale in Complex Ship Design*, SWZ Maritime 10(133), pp.31–37

DNV GL (2018), *Digital Solutions Brochure: Nauticus Hull*, DNV GL, Hovik

GAO, Y.; XIONG, Y.; GAO, X.; JIA, K.; PAN, J.; BI, Y.; …; WANG, H. (2023), *Retrieval-Augmented Generation for Large Language Models: A Survey*, ArXiv, abs/2312.10997

HMESHAH, K.; ERBACH, N.; BRONSART, R. (2015), *Impact of production requirements on high-quality ship product data models*, J. Eng. for the Maritime Environment, 230(3), pp.508-522

JORGE, J.; AGIS, G.; EBRAHIMI, A.; KEANE, A. (2018), *Quantifying the effects of uncertainty in vessel design performance - A case study on factory stern trawlers*, IMDC Conf., Helsinki https://www.researchgate.net/publication/325797359

KOELMAN, H.J.; VEELO, B.N.; SEPPÄLÄ, L.; FILIUS, P. (2024), *Closing the gap between early and detailed ship design models*, IMDC Conf.

LE POOLE, J.; DUCHATEAU, E.; HOPMAN, H.; KANA, A.A. (2023), *Development and experimental testing of a collaborative design rationale method for early-stage ship layout design*, Int. J. Naval Architecture and Ocean Eng. 15

LEWIS, P.; PEREZ, E.; PIKTUS, A.; PETRONI, F.; KARPUKHIN, V.; GOYAL, N.; … KIELA, D. (2020), Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, https://doi.org/10.48550/arXiv.2005.11401

MAVRIS, D.; DeLAURENTIS, D. (2000), *Methodology for Examining the Simultaneous Impact of Requirements, Vehicle Characteristics, and Technologies on Military Aircraft Design*, 22nd Congress of the Int. Council on the Aeronautical Sciences (ICAS), Harrogate

OPENAI; ACHIAM, J.; ADLER, S.; AGARWAL, S.; AHMAD, L.; AKKAYA, I.; … ZOPH, B. (2023), GPT-4 Technical Report, https://doi.org/10.48550/arXiv.2303.08774

PAPANIKOLAOU, A. (2018), *A holistic approach to ship design: Optimisation of ship design and operation for life cycle (Vol. 1)*, Springer

RIGTERINK, D. (2014), *Methods for Analyzing Early Stage Naval Distributed Systems Designs, Employing Simplex, Multislice, and Multiplex Networks*, University of Michigan

SIDDHARTH, L.; LUO, J. (2024), *Retrieval Augmented Generation using Engineering Design Knowledge*, In S. U. o. T. a. Design (Ed.)

SOMAN, R.R.; ANDRUS, M.; BOSWORTH, M.; LEONARD, I.; STEURER, M. (2015), *Approach to develop ship design evaluation rule-base*, 2015 IEEE Electric Ship Technologies Symp. (ESTS)

VoyageAI (2024), *voyage-multimodal-3: all-in-one embedding model for interleaved text, images, and screenshots*, VoyageAI

YLLIKÄINEN, M. (2019), *Effective utilization of digital design assets in post-design phases*, https://www.cadmatic.com/en/resources/articles/effective-utilization-of-digital-design-assets-in-post-design-phases/#:~:text=More%20controlled%20verification,the%20information%2C%20thereby%20improving%20traceability

Generative Algorithms in Early Ship Design: An Exploration of Hull Subdivision Generation

Diego De León, NHL Stenden, Leeuwarden/Netherlands, diego.de.leon.wug@nhlstenden.com
Herbert Koelman, NHL Stenden, Leeuwarden/Netherlands, herbert.koelman@nhlstenden.com

Abstract

This paper explores the potential for a data-driven tool to aid in the early ship design process, through the generation of subdivisions for the general layout via a proof-of-concept prototype which leverages a GAN to create plausible layout alternatives. The software implementation integrates a BSP tree structure for parametrisation, and a CAD geometry implementation. To work within the intrinsic limitations of generative algorithms, the decision-making is made by a naval architect, targeting facilitating the evaluation of multiple concepts and broadening the design possibilities. The paper describes the functioning of the proof-of-concept prototype, considerations on its creation and applicability.

1. Introduction

The current state of ship design is caught in between the rapid development of new computational technologies, and the challenges of a fundamental change in the industry propelled by environmental and legislative pushes towards decarbonisation and sustainability.

The diversity of solutions needed for sustainable propulsion and the implementation of energy saving technologies means that the new design processes should allow for a faster evaluation of multiple solutions, which is a change in paradigm from previous methodologies that had an immutable constant in their source of energy.

This multiple solution paradigm leads to the question of how to use these new computational technologies to enhance the ship design process. Within this project the proposed answer is the fast ideation at the beginning of the ship design process using generative algorithms, for the generation of multiple initial ship layouts as a base for naval architects and engineers to evaluate and work on, accelerating the initial process and allowing for the consideration of more possibilities.

2. Ship subdivision and layout rationale

The general arrangement of a ship plays a critical role in determining its functional and operational performance. In the early design stages, layout decisions establish the foundation for how spaces interact, how systems are integrated, and how future technologies, such as alternative propulsion can be accommodated. Despite this central role, the general layout remains one of the least digitally supported areas in the ship design process.

This gap is especially evident as the maritime industry shifts towards greener propulsion systems and more modular, adaptive vessels. Alternative propulsion solutions, often come with unique spatial and engineering requirements. Traditional design processes, relying heavily on expert intuition and manual iteration, struggle to efficiently explore the new design spaces that these technologies introduce. A tool that can rapidly generate and evaluate a wide variety of layout configurations becomes increasingly valuable in this context.

2.1. Energy transition challenges

The maritime industry has seen a series of changes in its long history as new technologies become available and offer more practical means of moving a ship. Multiple transitions mean multiple study cases that show how the industry and technological landscape has taken every change, *Herdzik (2023)* but certain parallels can be observed.

Every transition has been initiated by a change in technology, the availability of a new solution that supersedes the previously dominant technology due to practical, economical or technological reasons, creating a solution-driven change *DNV (2019)*. These historical precedents of changes in the industry differ from the one presented by the current decarbonisation challenge, where the urge to replace the dependency on fossil fuels lies not in the technical limitations of fuel oil itself, but in the external impacts it creates.

This problem-driven change presents a higher degree of uncertainty, as the proposed solutions are very application dependent, creating a new design paradigm where one of the constants assumed during the conception of the vessel becomes a variable to be evaluated.

3. Technological considerations and algorithm rationale

The integration of digital tools such as Computer-Aided Design (CAD), Computer-Aided Engineering (CAE), and simulation platforms has long been a cornerstone of modern ship design and engineering. These technologies enable detailed modelling, performance analysis, and iterative refinement across multiple domains: from hydrodynamics and structural integrity to machinery layout and stability, *Roh (2018)*. However, despite their established role, these tools are typically deterministic in nature: they operate under defined rules with limited capacity to adapt or generalize beyond their programmed scope.

This deterministic character, while essential for validation and certification, can impose practical limitations when exploring vast and complex design spaces or responding to emerging performance criteria. The process is often constrained by the need for explicit specification, sequential workflows, and expert interpretation, leaving little room for data-driven intuition, fuzzy logic, or emergent design discovery, *Gaspar (2018)*. Optimisation frameworks have addressed some of these challenges by introducing iterative refinement and goal-oriented processes, yet they still rely on predefined parameters and objective functions.

In contrast, data-driven approaches, particularly those enabled by machine learning, offer a complementary perspective. These methods shift the focus from calculating exact outputs to learning patterns, generalising behaviour, and uncovering structure from data. Where deterministic tools aim for precision and reproducibility, data-driven systems can support design exploration and design variation, enabling new forms of support in early-stage design and conceptual phase. Crucially, these approaches do not replace traditional engineering tools but rather extend their capabilities, bridging the gap between physical modelling and computational intuition.

3.1 Data as a resource

A critical issue is that most machine learning models, particularly supervised learning approaches, require labeled data to function effectively, *Huang (2024)*. This means that raw data must often be accompanied by annotations that indicate what it. Without such contextual labeling, data lacks the structure necessary for models to learn meaningful associations or make accurate predictions, *Markova (2022)*. The process of labeling data is often time-consuming and resource-intensive, particularly in domains that require domain expertise as marine engineering. As a result, the availability of labeled data can become a bottleneck in the development and deployment of machine learning systems.

In engineering and design contexts, including the maritime domain, data for machine learning applications originates from a range of sources. The two most common sources in practice are operational data and historical design data.

Operational data refers to information generated during the actual use of a system or product, such as sensor readings, performance logs, maintenance records, and environmental conditions. In maritime applications, this could include engine performance metrics, fuel consumption rates, route tracking, or structural responses under various sea states *deGeus-Moussault (2024)*. This data provides valuable

insights into how designs perform in real-world conditions, supporting tasks like predictive maintenance, performance optimization, and adaptive control.

Existing design data, derived from past projects and legacy systems, whether stored as CAD models, simulation results, or design tables, represent a repository of engineering knowledge and design intent. Machine learning models can use this information as a basis for pattern recognition, benchmarking, or generating new concepts inspired by proven solutions. This form of data reuse supports the notion of "learning from experience," enabling algorithms to build upon decades of accumulated engineering practice.

As an alternative to data from existing ships, synthetic data is artificially generated rather than collected from real-world observations. In engineering and design domains, this data is often produced using simulation environments or mathematical models that replicate the behaviour of physical systems under controlled conditions. Synthetic data serves as a valuable complement to real-world datasets, particularly in scenarios where empirical data is scarce, incomplete, sensitive, or expensive to obtain. However, synthetic data also comes with limitations, *Picard (2023)*. A key concern is fidelity, whether the synthetic data accurately reflects the complexity and variability of real-world phenomena. If the data does not capture important nuances, models trained on it may fail to generalize or may overfit to artificial patterns. Another issue is the potential for bias introduced by the assumptions or simplifications embedded in the generation process.

4. Prototype development

The first prototype developed in this project demonstrates a full pipeline for procedural ship layout generation based on a Binary Space Partitioning (BSP) tree structure. At its core, this prototype illustrates how early-stage layout decisions can be algorithmically generated, geometrically modeled, and evaluated, all within an open, modular framework designed for future extensibility.

4.1. CAD geometry background and BSP

To support the development of a generative layout prototype, the choice of a suitable computer-aided design (CAD) backend is a critical decision. In this context, OpenCASCADE offers a powerful toolkit for 3D modeling and computational geometry.

This solution is also particularly well-suited for implementing a Binary Space Partitioning (BSP) tree approach, which is central to this project's layout generation logic. BSP trees provide a structured way to recursively subdivide a design space into functional compartments, making them ideal for representing compartmentalized layouts such as ship interiors, *De Koning et all (2011)*. OpenCASCADE's geometric and topological modeling capabilities allow for precise and pragmatic creation, manipulation, and visualization of the partitions defined by the BSP structure. This compatibility enables seamless integration between abstract spatial logic and concrete geometric representation. Each node in a BSP tree can be directly mapped to a volumetric shape or compartment in OpenCASCADE, ensuring that layout generation remains both computationally efficient and geometrically meaningful. This synergy allows the prototype to transition smoothly between data-driven logic and engineering with a valid geometry, an essential feature for early design workflows that combine algorithmic generation with CAD integration.

4.2. Generative Adversarial Network

To address the challenge of generating plausible and diverse general layouts for ships, this project implements a Generative Adversarial Network (GAN) architecture. GANs are a class of machine learning models particularly well-suited for generative tasks, with some existing applications within ship design, *Khan (2023)*.

At the core of a GAN are two neural networks with opposing goals: the generator and the discriminator. The generator attempts to create candidate layouts that mimic real designs, starting from random input (noise) or guided design parameters. The discriminator, in contrast, evaluates the validity of the generated layouts by either comparing them to a dataset of real examples or using other types of filters. The two networks engage in a zero-sum game: as the generator improves its ability to fool the discriminator, the discriminator simultaneously becomes better at detecting synthetic designs. This adversarial training dynamic drives both networks to improve continuously, resulting in progressively higher-quality generated outputs, *Cresswell (2018)*.

A key advantage of this approach lies in its modularity and adaptability. The generator and discriminator can be developed, trained, and adjusted independently. This flexibility allows to explore a variety of architectures, learning strategies, and input features without rebuilding the entire system.

The discriminator is not limited to being a single neural network. The modularity of the model allows the exploration of the use of non-neural, computational discriminators. Existing engineering tools can act as evaluators that provide a judgment of generated layouts. This approach aligns well with the hybrid nature of early-stage design, where domain knowledge and engineering logic still play a crucial role. By allowing the discriminator to include engineering mathematical models, simulation tools or rule-based validators, the system gains a significant advantage in producing realistic, and also functional and constraint-compliant designs. This type of evaluation also facilitates the user to enter specific types of constraints to obtain the desired output.

Another important benefit is the versatility of the model framework. Both networks can be replaced or enhanced with other machine learning methods, such as autoencoders, reinforcement learning agents, or decision trees, depending on the goals of a specific design task. The GAN framework becomes a flexible experimental sandbox in which new layout strategies can be tested and improved iteratively.

The use of a GAN allows this prototype to go beyond rule-based generation by learning design patterns directly from data. This makes it possible to support early-stage designers not just with static templates, but with dynamically generated layouts that respond to learned design preferences and can evolve through training. Combined with a human-in-the-loop workflow, this approach opens the door to a powerful new class of design tools that support both automation and expert oversight.

5. Implementation and workflow

A design input is expected from the naval architect utilising the tool and from the engineering files, at this stage the .iges file. For this proof of concept prototype the input expected includes the ship dimensions, basic operational requirements such as expected minimal range and desired cargo volume, and basic expected design decisions to test, such as the fuel type to evaluate. These required designs will be subject to change depending on the implementation of data exchange with other software tools within the platform and future capabilities of the requirement and engineering check modules of the discriminator.

The process begins with the random generation of a BSP tree, which defines a hierarchical spatial subdivision of a cubic design volume. This volume represents the internal space of the ship, abstracted to allow flexible partitioning without yet being constrained by the hull shape. The generator is currently driven by random number generators but it is designed to be replaced by a neural network in future iterations, enabling better data-driven partitioning strategies.

Once the BSP structure is generated, it is mapped into a set of 3D compartments using OpenCASCADE's solid modeling tools. The resulting subdivided solid serves as the first stage of the layout representation. To enforce geometric realism, a Boolean subtraction operation is performed using a given hull form. This operation trims the subdivisions to fit within the available internal volume of the vessel, ensuring that generated layouts remain within feasible spatial bounds.

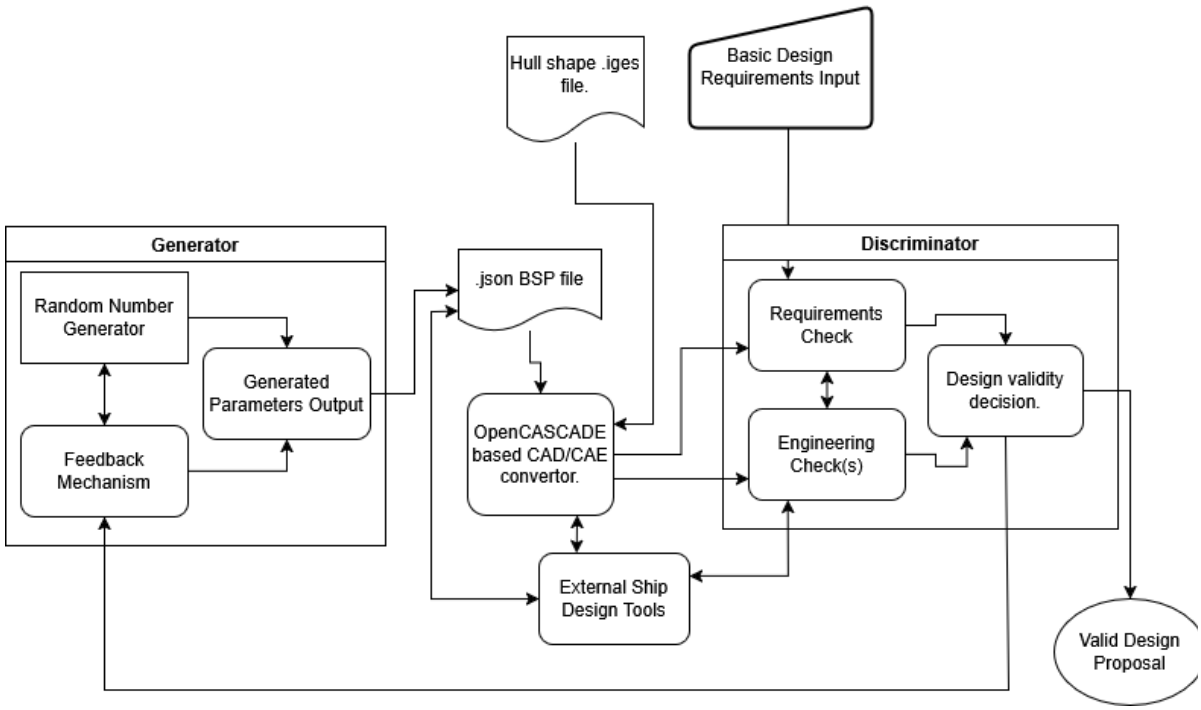


Fig.1: D^3: SEA general functioning principle

Following this, each compartment is evaluated in terms of position, size, and volume, providing a basis for early-stage layout analysis. These metrics serve as feedback to the generator, forming a simplified form of discriminator logic. The discriminator, in this initial prototype, applies a set of rule-based and numerical filters, such as minimum compartment volume or geometric alignment, which help identify and discard invalid or poorly formed layouts. This first step sets the stage for future improvements where discriminators may incorporate historical layout data, engineering rules, or even existing digital tools, including those being developed in the EU Horizon SEUS platform.

All layout data, including the BSP tree and resulting geometry, are stored in a .json format, offering human readability, logging, and integration with other systems such as SARC PIAS subdivision tools. For CAD interoperability, the OpenCASCADE architecture the prototype is built upon also supports export in .STEP and .IGES formats, ensuring compatibility with standard engineering workflows and tools. This flexibility positions the prototype as a contributor to the broader SEUS digital platform, enabling downstream integration with other tools for evaluation, simulation, or visualization.

Where each subdivision is described by the plane to which it is parallel and the fraction of the volume at which it happens. The base subdivision continues through a "branch" divided by right and left paths of the BSP tree until it finishes on a "leaf", where extra information can be added, in this case volume and the intended use of the subdivision, for engineering use.

6. Results and analysis

The first experiments with the prototype were carried out in three stages. In the first stage, BSP trees were generated without applying any constraints, producing purely random subdivisions within the design volume. These results illustrated the baseline behaviour of the generator, but also confirmed that completely unconstrained subdivisions have limited value even for early-stage ideation. In the second stage, a constraint was introduced to ensure a prescribed volumetric balance between fuel and cargo spaces. In the third stage, a minimum deck height constraint was added alongside the volume requirement; this configuration was tested but not evaluated in depth within the current work.

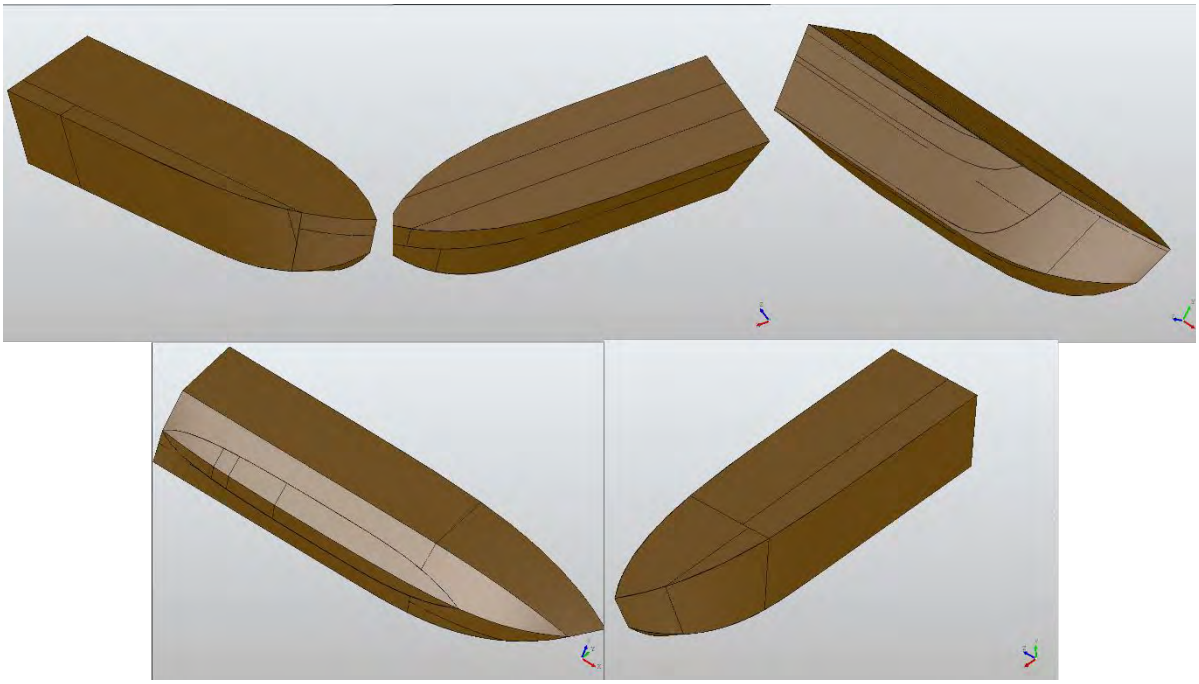


Fig.2: Simple volume distribution filter based subdivisions

The layouts generated at this stage remain preliminary and abstract. While they comply with the imposed geometric filters, they do not yet exhibit the coherence or functionality expected of practical ship layouts. In their current form, the outputs are not directly applicable to engineering use. This distinction underlines the difference between algorithmically generated results and layouts designed by human labour. A naval architect would not only respect geometric and volumetric constraints, but would also incorporate a wide range of implicit, "common sense" considerations. For example, simple concepts such as a basic level of symmetry, which is a common feature in ship design due to both aesthetic and practical considerations, are not accounted for in this version of the algorithm. These layers of design logic are difficult to capture through purely numerical filtering and highlight the gap between algorithmic feasibility and professional design practice.

The experiments also reveal the inherent complexity of internal subdivision in ship design. Effective compartmentalisation is a multi-variable problem, where while decisions can be reduced to a set of numerical rules as by engineering practice, a complete systematic analysis would require a very high number of considerations. Although hard constraints provide a necessary foundation, the level of complexity may benefit from more abstract reasoning. For this reason, the use of existing subdivision data as reference patterns emerges as an important complementary strategy. By learning from established examples, the algorithm may approximate more complex design rules that cannot easily be formalised, exploiting the known advantages of data-driven approaches.

Another observation is that the prototype is not intended as an optimisation framework. Traditional optimisation methods target performance measures and operate within fixed sets of constraints and objectives. In contrast, the present approach seeks to support faster ideation at the earliest design stages. Its value lies in the capacity to generate multiple plausible alternatives quickly, enabling designers to evaluate a broader range of possibilities than would be practical through manual iteration alone. In this sense, the prototype is positioned as an augmentation of human design capabilities, not a replacement for them.

While the results confirm the feasibility of BSP-based generative subdivision and demonstrate the potential of the approach as a tool for rapid concept generation, they also highlight the necessity of additional layers of evaluation logic, incorporation of reference data, and sustained human oversight. The tool's role is not to deliver final or optimised layouts, but to accelerate the ideation process and augment the designer's capacity to explore multiple pathways in the early phases of ship design.

7. Conclusions and future work

The prototype demonstrates the feasibility of using BSP trees for algorithmic generation of ship subdivisions and validates the approach as a foundation for further development. Its primary contribution is not the delivery of optimised or directly applicable layouts, but the demonstration of a generative framework capable of producing rapid concept alternatives. By accelerating early-stage ideation, such tools can augment the work of naval architects, providing a broader range of candidate layouts to consider at the outset of the design process.

The results also confirm that human supervision remains indispensable. While the generated layouts are geometrically consistent, they lack the context, experience and other implicit "common sense" knowledge that human designers naturally apply. This reinforces the need for a human-in-the-loop workflow in which the algorithm acts as a generator and the designer as evaluator and decision-maker.

Future development will focus on three complementary directions. First, the integration of external software solutions, already part of the broader project platform, offers a path to rapidly extending the set of evaluative filters available. By drawing on established tools rather than developing every component in isolation, it becomes possible to considerably raise the quality and realism of the generated results. Increasing the number and sophistication of filters is expected to directly translate into more coherent and practically relevant layouts, within reason and the capabilities of the tools to explore.

Second, the inclusion of training data from existing subdivisions will enable the system to move beyond purely numerical constraints. Learning from historical or reference layouts allows the generator to approximate the complexity of design logics in a similar way that human experience serves as a shortcut to the complex engineering mathematics in the design process. However, one of the main limitations in this direction is the availability of subdivision data in formats that are suitable for direct training. Overcoming this barrier will be essential for fully leveraging the potential of data-driven approaches, and potential approaches exploit the use of synthetic data and machine-learning-based data tagging and extraction tools.

The prototype establishes a solid foundation for BSP-based generative subdivision and demonstrates its potential as a tool for accelerating ideation in early-stage ship design. Its further development will depend on enriching the discriminator through integration with external software and embedding knowledge derived from existing subdivisions, ultimately enabling a hybrid approach where computational generation and human expertise work in tandem.

Acknowledgements

With strong gratitude to the EU Horizon SEUS project and its partners: NTNU, NHL Stenden, UTU, SARC B.V., CADMATIC, Gondan Shipyards & Ulstein Shipyards. To the European Union as the main sponsor of the project.

References

CRESWELL, A.; WHITE, T.; DUMOULIN, V.; ARULKUMARAN, K.; SENGUPTA, B.; BHARATH, A.A. (2018), *Generative adversarial networks: An overview*, IEEE Signal Processing Magazine 35/1, pp.53–65

DE GEUS-MOUSSAULT, S.; SEUBERS, H.; LINSKENS, H.; CORADDU, A.; PRUYN, J. (2024), *Operational Data for Sea Margin Calculations in Early Ship Design*, 15th IMDC Conf., Amsterdam

DE KONING, D.; KOELMAN, H.J.; HOPMAN, H. (2011), *A novel ship subdivision method and its application in constraint management of ship layout design*, 10th COMPIT Conf., Berlin, pp.292–304

DNV (2019), *Comparison of alternative marine fuels*, https://sea-lng.org/wp-content/uploads/2020/04/Alternative-Marine-Fuels-Study_final_report_25.09.19.pdf

GASPAR, H.M. (2018), *Data-driven ship design*, 9th COMPIT Conf., Pavone

HERDZIK, J. (2023), *Marine Fuel From The Past to The Future*, Scientific Journals of the Maritime University of Szczecin, https://repository.am.szczecin.pl/handle/123456789/2780

HUANG, Q.; ZHAO, T. (2024), *Data Collection and Labeling Techniques for Machine Learning*, https://arxiv.org/abs/2407.12793

KHAN, S.; GOUCHER-LAMBERT, K.; KOSTAS, K.; KAKLIS, P. (2023), *Shiphullgan: A generic parametric modeller for ship hull design using deep convolutional generative model*, Computer Methods Appl. Mech. Eng. 411

MARKOVA, V. (2022), *Impact of Data Preprocessing on Machine Learning Performance*, Academia.edu, https://www.academia.edu/67070662/Impact_of_Data_Preprocessing_on_Machine_Learning_Performance

PICARD, C.; SCHIFFMANN, J.; AHMED, F. (2023), *DATED: Guidelines for Creating Synthetic Datasets for Engineering Design Applications*, arXiv preprint arXiv:2305.09018

ROH, M.-I.; LEE, K.-Y. (2018), *Computational Ship Design*, Springer

VEELO, B.N. (2004), *Variations of Shape in Industrial Geometric Models*, PhD Thesis, NTNU, Trondheim

AI-Assisted Misalignment Detection in Technical Ship Drawings

Magnus August Thorsrud Weidemann, VARD, Ålesund/Norway, magnus.weidemann@vard.com
Henrique Murilo Gaspar, NTNU, Ålesund/Norway, henrique.gaspar@ntnu.no
Håvard Vollset Lien, VARD, Ålesund/Norway, havard.lien@vard.com

Abstract

This paper investigates the use of commercially available artificial intelligence for detecting misalignment errors in 2D technical ship drawings. Existing multimodal large language models were tested for their ability to interpret and analyze engineering drawings without needing to use custom coding or programming. A stepwise method was developed, evolving from initial tests on simple shapes to case studies with real general arrangement drawings from NTNU's research vessel Gunnerus and a Vard platform support vessel. Results show that, with tailored prompts, AI can reliably identify linear misalignments, offering potential as an assistant for naval architects. Though, current limitations require further development on the usability of the methods in an automatic way, as well as improvements in the AI tool and prompt.

1. 2D Drawings in Ship Design and Construction as Source of Errors and Revisions

The life cycle of a ship can be divided into multiple different phases. The development process is a major part of the ship life cycle and encompasses many of the early phases, such as concept development, engineering and final design, Fig.1.

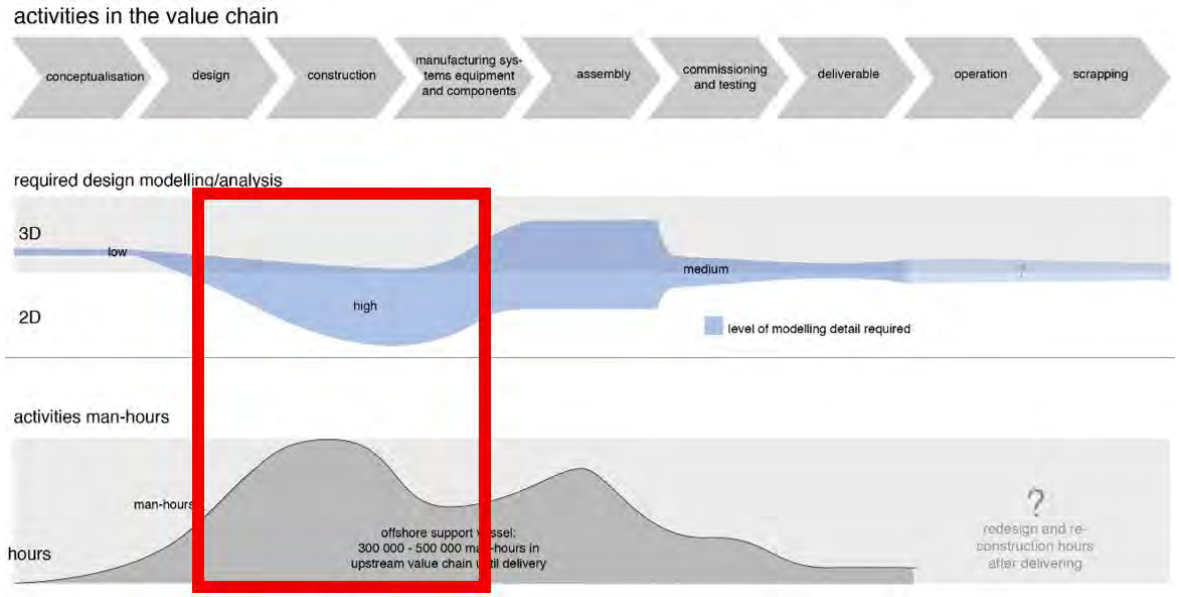


Fig.1: Activities and processes for the ship design value chain. The red box highlights the shift towards predominantly 2D drawings and increased number of man-hours spent on these phases, *Gaspar (2016)*

Ship design and construction players produce a large amount of 2D drawings when developing a ship. Most drawings are made individually in CAD tools by humans, meaning errors can occur. If errors are not detected early, it can negatively affect a project, leading to time and money being spent on redrawing and correcting. If errors are not identified and the ship is built with flaws, it can cause stresses not accounted for in the design phase and lead to a shortened useful life or a catastrophe in worst case.

Revisions are usually performed manually, both by the ship design office, shipyard classification societies, meaning valuable working hours are not spent on advancing the project. Mistakes discovered must be corrected and the new drawings are then re-checked, adding to the total cost.

In this context, we decided to investigate if the modern commercially available AI tools can be used more successfully in maritime design and engineering than pervious investigations, *Gaspar and Bertram (2023), Gaspar et al. (2023)*, specifically on how AI can assist naval architects in detecting errors in technical 2D ship drawings.

Misalignments in 2D CAD drawings, more specifically errors in straight vertical and horizontal elements in general arrangements are the focus of this paper. A misalignment is defined here as a line segment which in real life should make up, or be part of, a continuous line, but is not.

2. AI Tool Potential

AI is evolving fast, and innovations are happening continuously. It is becoming a useful tool in many industries with everything from customer service bots to autonomous cars. One of the advantages with AI is its ability to handle and process large amounts of digital data. It is therefore logical to investigate the practicality of using AI in ship design processes, considering much of the documentation is digital, including 2D technical drawings. If AI can help reduce the workload for naval architects without losing accuracy, economic gains are potential while ensuring the same quality. As the purpose of the paper is to utilize existing AI tools available, no programming or coding has been performed.

One of the most advanced systems using AI to detect drawing errors is a model using deep learning to detect common engineering errors in piping and instrumentation diagrams. The task is to perform quality checks of complex technical drawings for large petrochemical installations, Fig.2. The study performed a cost analysis confirming the potential for savings due to reduced manual checks, *Dzhusupova et al. (2023)*.

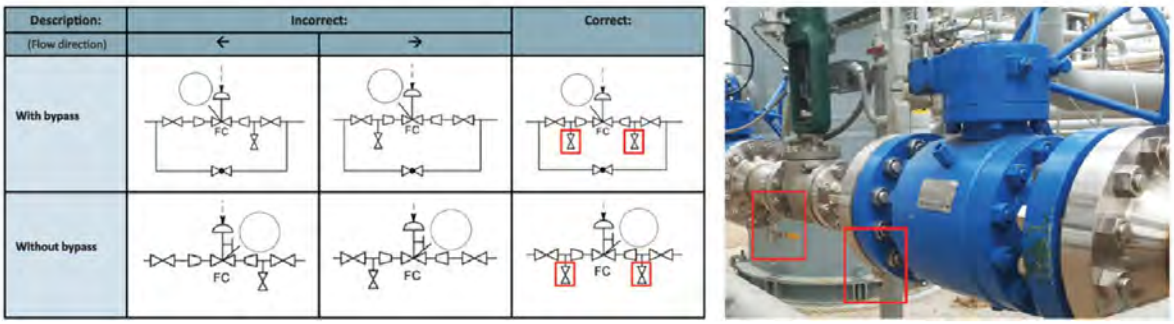


Fig.2: Example of the type of errors the program is trained to detect, *Dzhusupova et al. (2023)*

Considering large language model chatbots are some of the most user-friendly and accessible AI tools existing on the market, it was decided that LLMs would be used for the research and development of an error detection method. The free version will be tested first. If results are promising, but the output limit is consistently reached, the paid subscriptions will be used. This is to facilitate a steadier workflow and potentially exploit better image analyses features. Chosen AI programs were ChatGPT, Gemini, Deepseek, Qwen, Claude, Microsoft Copilot and Aicado.ai.

3. Methodology

3.1 Type I and Type II Errors

We decided to investigate our conception of “error” using the basic hypothesis testing types: Type I and II. Type I error is understood to be when the hypothesis is approved, but is in fact incorrect and should have been rejected. Type II error mistakenly rejects the hypothesis. See Table 1 for the hypothesis case of this paper. For this project, the preferred outcome is to have Type I errors. This is because no markings on misaligned lines mean errors will go undetected.

Table I: The two error types in the context of this paper's objective

		There is a misalignment	
		True	False
AI misalignment reporting	Misalignment exists	Found misalignment	Reported a correct line, Type I error
	Misalignment does not exist	Did not report misalignment, Type II error	Found no misalignment

3.2. Testing Procedure

The initial testing, covering processes 1, 2 and 3 and decisions A, B and C from the flow chart in Fig.3, starts with every AI program previously mentioned (ChatGPT, Gemini, Deepseek, Qwen, Claude, Microsoft Copilot and Aicado.ai) getting identical input questions and a general arrangement PDF. Each program's potential will be determined by how well it can understand and interpret the overall content of the technical drawings. Based on their ability to produce output with accurate information of ship elements and area locations, a decision will be made regarding whether to continue with PDF or switch to PNG format. If an AI model demonstrates poor initial understanding or produces incorrect results, it will be removed from further testing.

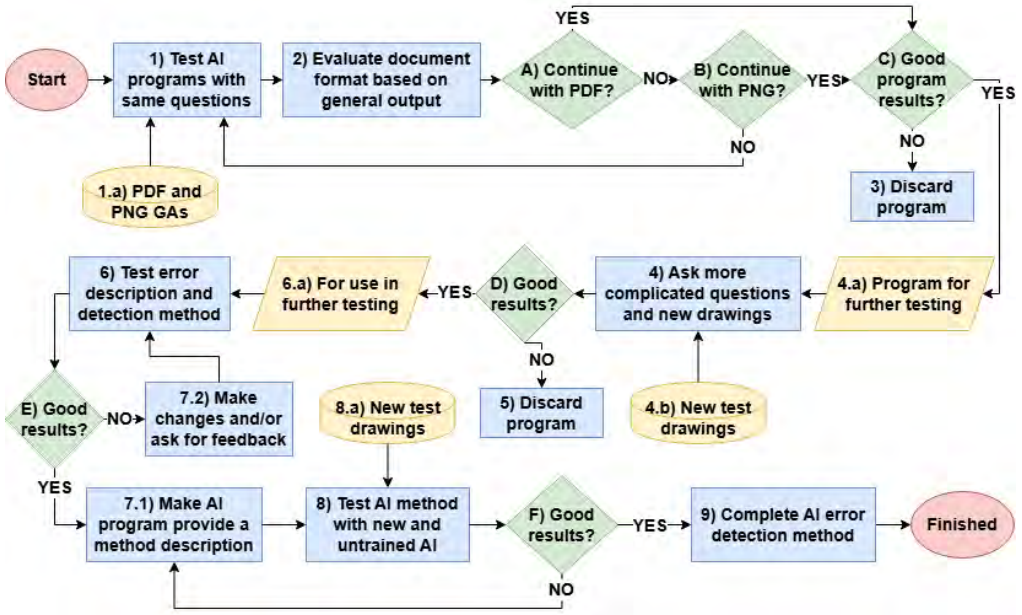


Fig.3: Flow chart illustrating the testing and development method

In the second phase, the remaining tools will be asked more specific and detailed questions, testing their technical accuracy and misalignment detection. This includes increasing the difficulty of the tasks and introducing different designs. The AI tools will be assessed by their output quality, and for how they improve with more detailed prompt instructions. Underperforming programs will be discarded.

The final phase will use the AI tool that best performed (ChatGPT) to help create the error detection procedure. The programs will be prompted to help make definitions in their own words which an untrained AI can understand and use to produce the same output. To validate and confirm the repeatable result output, the method will be tested again using new drawings and untrained conversations. Final evaluations will focus on accuracy, consistency and how easily the method can be applied in real-life.

3.2. Initial Testing and evaluation

A PDF with the general arrangement from *Eide and Weidemann (2023)* was tested first. The output was not good enough when asked to locate different elements. To potentially increase the accuracy, the

drawing was cropped to only show the inside profile view. This did not provide improvements, and areas of interest were highlighted to help guide the AI, Fig.4. The programs were prompted to identify which frames the bulkheads are located on, here marked in yellow and red respectively.

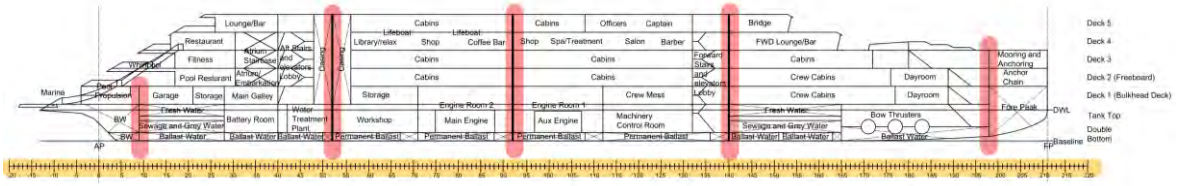


Fig.4: Side profile of bachelor general arrangement with highlights

However, the results for all of them were not good and two programs did not take PDF. The decision was made to convert the PDFs to PNG and test. This improved the results for a few programs. DeepSeek, Qwen and Copilot were the least accurate and discarded from the testing. Janus Pro and Aicado did not provide the most accurate results either. Output was not saved after a new prompt was given. They were discarded as well. The paid subscriptions of ChatGPT Pro and Claude Pro were used as the generator limit was reached. To remove the possibility of using text to guess the location of areas or for counting, and to reduce the amount of detail to analyze even more, a new simple ship side profile was drawn, Fig.5. It only contains the main structural elements. To provide an idea of the distances relevant for the error detection, minimum and maximum misaligned measurement marks were added to the left under the keel.

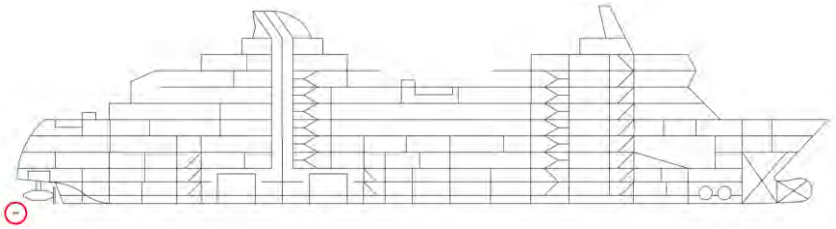


Fig.5: Simple ship profile general arrangement with minimum and maximum error distance measurements in the red circle

The first question asked was "What do you see in this image?". Most LMMs gave correct responses even though the image does not contain any text that could inform them. This proves that they can interpret PNG image content. Multiple images were made with different misalignment errors for the AI to try and detect. The AI was provided with prompts explaining the misalignment definition. The detection was inaccurate, and the misalignment marks did not provide any increased accuracy. A grid was thought of to be used for location conveying. However, the grid did not convey element location any better and was thus removed as well.

Table II shows a summary of the initial testing. Information about what kind of input the different AI programs were able to process, with additional comments about their limitations, is provided. Green indicates that the tool can manage the file format or is accurate, yellow means medium accuracy and red indicates that the format is not accepted or that the program is inaccurate. The accuracy evaluation is based on the student's experience from tests performed in February 2025. Changes may have been made since then, which can potentially lead to different output today.

Table II: Capability and accuracy of the AI programs

Program	AI type	PDF	PNG	General accuracy
ChatGPT	LLM	Only able to extract text		
Gemini	LLM	Only able to extract text		
DeepSeek	LLM	Only able to extract text	Only able to extract text	

Janus Pro	Computer vision		Answers are not saved	
Qwen	LLM	Only able to extract text		
Claude	LLM	Only able to extract text		
Copilot	LLM	Only able to extract text		
Aicado	Computer vision		Answers are not saved	

3.3. Pixel Testing: Patterns for Measuring and Shape Recognition

Even though most of the LLMs indicated that they could interpret the technical drawings, no detailed prompts explaining how misalignments look like and how to find and report their position, managed to create the correct output. A decision was made to change the approach and try using pixel analyses instead. The idea was to have drawings with one-pixel wide lines and ask the AI programs to state which rows and columns misalignments have occurred in. Even with the approach change, the next steps cover steps 4 and 5 and the decision D from the flow chart in Fig.3.

We decided then to create patterns, limiting first to input simple shapes, 50 by 50 pixel images in size, with one-pixel wide lines. The three remaining programs, ChatGPT, Gemini and Claude, were all given the first image and prompted to state the resolution and the number of black pixels. ChatGPT managed to answer both questions correctly after a few tries. Claude's output informed of its inability to determine image resolution. Gemini repeatedly gave the incorrect number of black pixels. This led to them to being removed from the rest of the testing. The continued testing covers the remaining processes and decisions shown in Fig.3's flow chart. From the first test images, the importance of columns and rows structuring was introduced. The discovery that ChatGPT also reverses the colors was made. Once these misunderstandings were sorted out, it was clear that the AI tool can provide accurate and reliable answers.

It was revealed that ChatGPT uses another coordinate system than the cartesian one. The answers were close, but one or two pixels off. ChatGPT uses the default image processing code. Columns are counted left to right, but rows are counted from top to bottom. The values also start at 0. This helped clarify prompts and specify descriptions. ChatGPT was instructed to use the cartesian coordinate system with origo in the lower left corner and start value as one.

During Pixel test 8, ChatGPT was asked to provide the length of the lines in the image. It consistently provided a length that was one pixel too long for both horizontal lines. Therefore, it had to create an image highlighting the two horizontal lines, one in alternating red and blue pixels and the other in alternating green and yellow. From Fig.6 we can see that it includes all black pixels on the same row. This illustration informed of how precise the information must be and helped to further improve the prompt engineering. ChatGPT was informed that a line in these tests is defined as two or more black pixels connected by the edges.

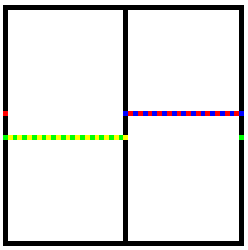


Fig.6: Horizontal lines highlighted with alternating colors

When asked to count the distance from the border to the center square in "Pixel test 14" it struggled to complete the task. Fig.7 shows how only one pixel on each border line got colored in. The last error could be caused by the previous explanation that only connected black pixels count as lines. White

spaces interrupt lines, but this is not a concern when measuring distance from one part to another. After altering the input, ChatGPT found the right answer, providing an image showing the proper distance measurement, Fig.8.

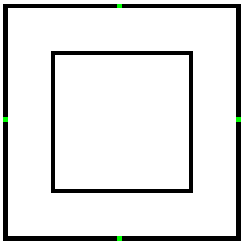


Fig.7: Example of wrong distance measurement

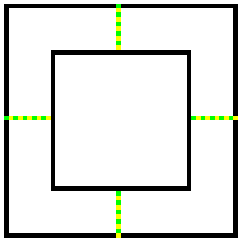


Fig.8: Correct distance measurement

Because of the struggle to produce the correct result, ChatGPT was prompted to give feedback on the prompts used. Some useful answers included defining words better, such as the word "distance" and specifying whether black pixels from other lines should be included or not. This started to produce better results.

As most answers started to improve, larger 100 by 100 pixel images were tested. ChatGPT was able to measure distances and provide image property details accurately. Fig.9 for an example. It also understood layer-properties, as was demonstrated with “Pixel test large 4”, Fig.10. It can even color in different elements of an image Fig.11 for a colored quarter circle. Though, it struggled when the shapes became too complicated and intertwined. Fig.12, which is output from a 200 by 200 pixel image.

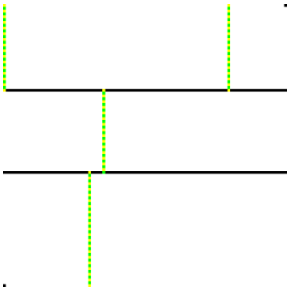


Fig.9: Vertical line segments highlighted

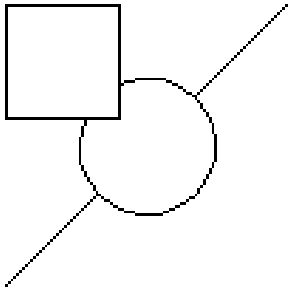


Fig.10: Pixel test large 4

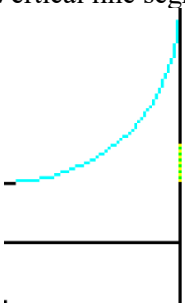


Fig.11: Distance highlight and color change.

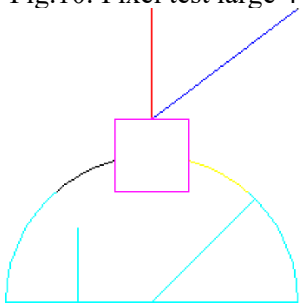


Fig.12: Attempt at coloring in each composite shape in a unique color.

As the initial plan was for ChatGPT to provide column and row number of where errors are located, the idea of highlighting errors was thought of due to the proven ability of coloring. Thus, the next image contained many shapes in a random grouping. The first prompt was to color in every circle red, only color the border of each square blue and the left half of each triangle border green. The output had no flaws, Fig.13. The next task was to measure the distance from the center square borders to the nearest side of each remaining square. This output was also correct, Fig.14. Consequently, both alternatives are viable, but highlighting was chosen as it would be the simplest inspection method for a human.

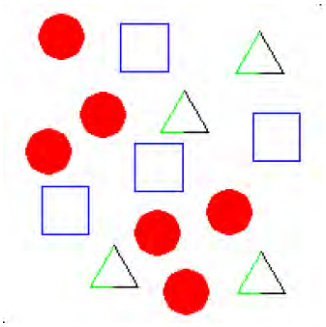


Fig.13: Correct color changes

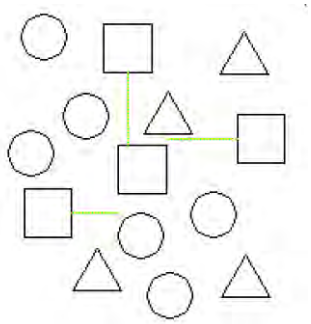


Fig.14: Correct distance measurements

“Pixel test larger 3” has a 6 by 6 square grid. This test was created with the purpose of developing the proper error detection method. It has multiple misalignments of different sizes and orientations. A simple description of the error type was provided and an output highlighting the misalignments was requested. Unfortunately, every line was highlighted, Fig.15. After working back and forth with the error description and trying multiple tactics, the results still colored every line. An idea was to give the location of each continuous and misaligned line. ChatGPT colored correct lines green and error lines red, Fig.16. This could help it understand what an error looks like compared to correct ones. When questioned why the detection failed earlier, it expressed that the description was too generalized and sensitive.

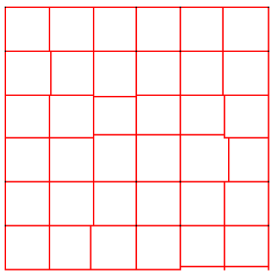


Fig.15: Incorrect highlight of every line

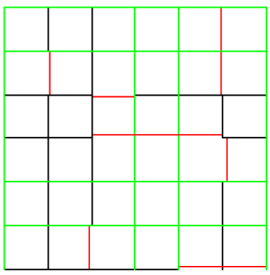


Fig.16: Correct marking after locations were provided

ChatGPT made a new description. To make it more specific, a search radius was provided which states that the AI should search for parallel fragments within a certain area. 10 pixels were chosen as the search radius. To test the effectiveness of the latest method description, new images were used. The first ones only contain one misalignment. This was to make it easier to understand what ChatGPT struggles with when trying to locate and detect errors. Later images gradually got more, with errors going in both directions and breaking elsewhere than at the intersection points as well.

The new description continued to need improvements. A new rule was created saying that pixels on the left or right side of black pixels in a vertical line can be ignored until a break in the line is found. The same goes for horizontal lines where pixels above or below the line can be ignored. However, this became too strict, and errors mistakenly required broken connections on all sides of the line.

A correction was made where it would check broken top and bottom connections for vertical lines and broken left and right connections for horizontal lines. The new analysis marked every line except for the border lines, Fig.17. The general location and description of the vertical error was given, and the new analysis gave an image where every horizontal line got marked, Fig.18.

ChatGPT explained that during the scanning of each column for the vertical error, pixels belonging to horizontal lines were flagged. This is because each black pixel is considered a line and from a vertical point of view, as they all had white pixels connected to the top and bottom edges. ChatGPT was reminded that a line is two or more pixels long.

The new rule was implemented, and the next test provided the first correct output. ChatGPT managed to detect a misalignment and mark it red, Fig.19. Now that a result had provided promising output, the method had to be evaluated several times to verify the results. Fig.20 shows the correct result of the last and most difficult grid.

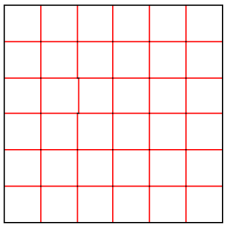


Fig.17: Every line highlighted except border lines

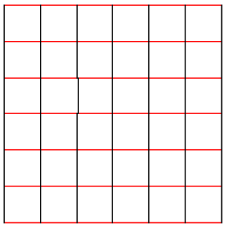


Fig.18: Every horizontal line incorrectly marked

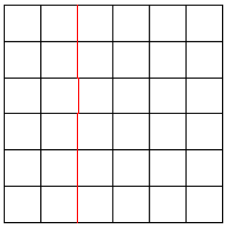


Fig.19: First correct error marking

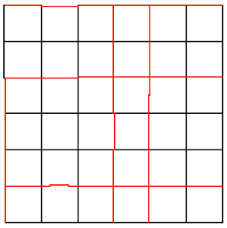


Fig.20: Marked errors from the last grid

3.4. Pixel Testing: Simple Technical Ship Drawings

The detection method used for the grid was verified. Next step was to prove that the method could work with different and larger, more complicated ship drawings, changing the description if necessary. A 400 by 200 pixel image of a ship profile was made. Multiple versions were made, each of them having different misalignment errors.

In the first tests, parts of curved and angled lines were marked. This is because these lines are made up of smaller straight line segments when the thickness is only one pixel wide. The min_line_length setting was created and increased to 6 to filter out smaller elements making up these shapes. The search_radius was set to 4 pixels. This worked and all misalignments were highlighted red, Fig.21. However, considering the complexity of the outline, some straight line segments from angled and curved lines still fell within the misalignment definition and were marked. This is a Type I error, which is preferable. Final validation was made with three other ship profile drawings, increasing the size to 800 by 200, Fig.22. The output was the same, confirming the findings from the previous tests.

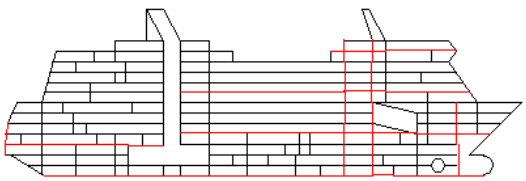


Fig.21: Correct ship error marking

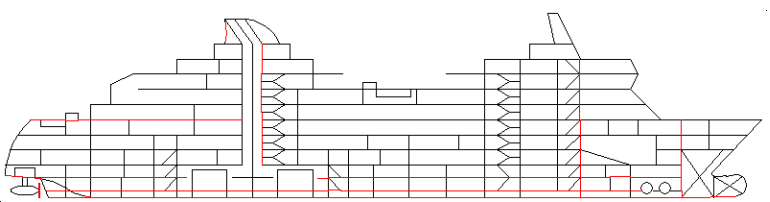


Fig.22: An example of a result from the large technical ship drawing testing

3.5. Limitations and Explanations

When investigating image size limits, the maximum resolution was found to be 2048 by 248 pixels. In real life scale, one pixel will represent one measurement unit. The limitation means only smaller areas can be analyzed per image. This is apparent for ship drawings, which often use millimeter units, and ships are larger than 2 by 2 meters. Therefore, multiple images must be made to cover all the required area. Overlap between the images is also important so to not lose errors close to the edge of the images.

The current AI-based detection method is restricted to detect misalignments of straight, vertical and horizontal lines. This limitation was a purposeful decision to help make the initial development goal simple. By focusing exclusively on straight elements and error descriptions with parameters, the prompts and evaluation criteria could clearly be defined. It also helps in determining the AI's responses and measuring its performance. However, this also means the method is not yet suitable for identifying more complex or varied types of errors that are common in technical ship drawings.

There are three parameters in the analysis which the user must specify. min_line_length is the minimum number of pixels that make up what is a straight line. search_radius is the maximum number of pixels away from the line the analysis will investigate for new line segments. These two settings affect what size of errors are to be detected and can be adjusted for optimal precision. image_resolution_scale will increase the resolution of the output image by the value provided. This setting is useful for smaller input images, as it will make the manual inspection easier.

4. Case Studies

4.1. Gunnerus General Arrangement Assessment

General arrangements from real ships were used as well, considering they are more detailed and will test the method in a real-life scenario.

The first real general arrangement to be used was of NTNU's research vessel Gunnerus. Sections of deck A were chosen to be tested first as it looked like a good mix between details and general structure. Test images were made, but without overlap as the main errors were handmade and known.

The output was good, as every purpose-made error got detected, lines from more detailed elements were also marked if they fit the error settings. Fig.23 shows the result after using a minimum length of 10 pixels and a search radius of 15 pixels. Using a minimum length of 30 pixels and search radius of 5 pixels detected less, as can be seen in Fig.24. One can also see how the complete faucet is colored red in Fig.23 while partially colored in Fig.24. This is because solid-colored elements are in theory made up of black parallel lines and the method detects these lines as errors. When the lines in the colored elements are too small, they fall outside the detection criteria.

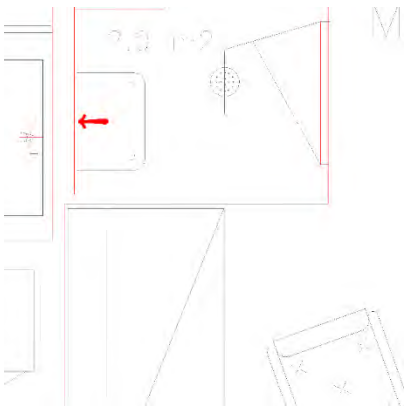


Fig.23: Error detection using a line length of 10 and radius of 15

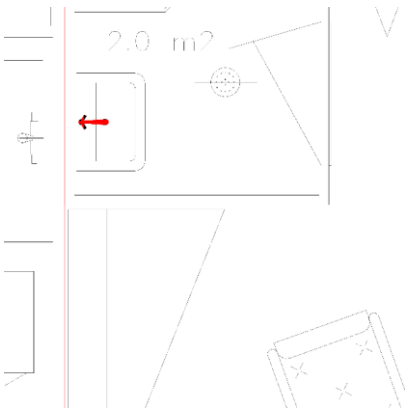


Fig.24: Error detection using a line length of 30 and radius of 5

4.2. Vard PSV General Arrangement Assessment

Another general arrangement was provided by Vard Group AS. The ship is a platform support vessel, PSV. The drawing was examined to locate errors which ChatGPT could detect in an analysis. A few errors were found, Fig.25. However, these are different than what is currently intended for the detection created method. A few misalignments were therefore made to the structure in the bow area of the ship. Fig.26 for misalignments marked in red. Fig.27 shows the bow region, and the green box marks the selected search area.

Fig.25: Examples of misalignments not covered by detection method

Fig.26: Four purpose-made misalignments, colored red for visibility

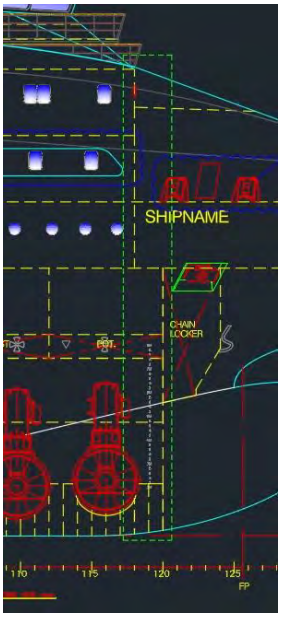


Fig.27: Bow area with green dashed box indicating desired search area

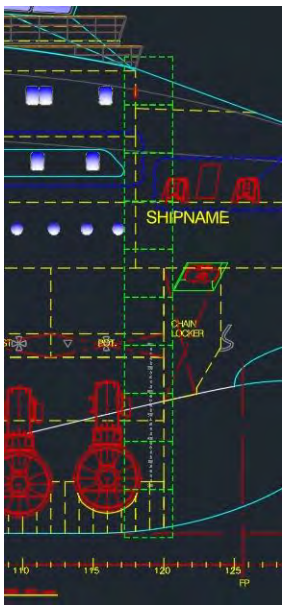


Fig.28: Green dashed boxes indicate print windows for image creation

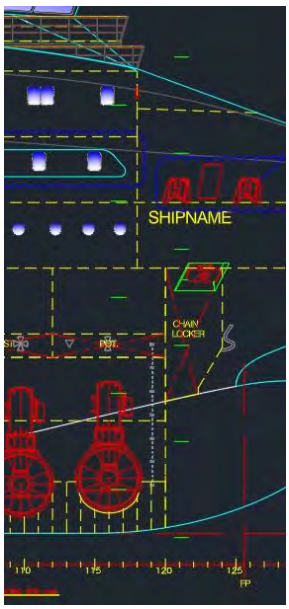


Fig.29: Small guidelines extending from removed image boxes

Fig.30: All lines made solid with thickness of 0.0 mm

After finding the desired search area, squares with a size of 2048 by 2048 mm were placed, outlining where each image should be made. Fig.28. To not include the box-outline in the images, they were

replaced by smaller lines at the corners, Fig.29. These are used as guidance when selecting the window in the print settings. The last step was to ensure that all lines are solid and have a thickness of 0.0 mm, which will make them 1 pixel thick, Fig.30.

10 images were required to cover the total height of the selected bow section. For a proper analysis, more images would be needed. No overlap was made with these images as well. Fig.31 shows one of the images containing a marked misalignment. The analysis was performed using a min_line_length and search_radius of 300 and 5, respectively.

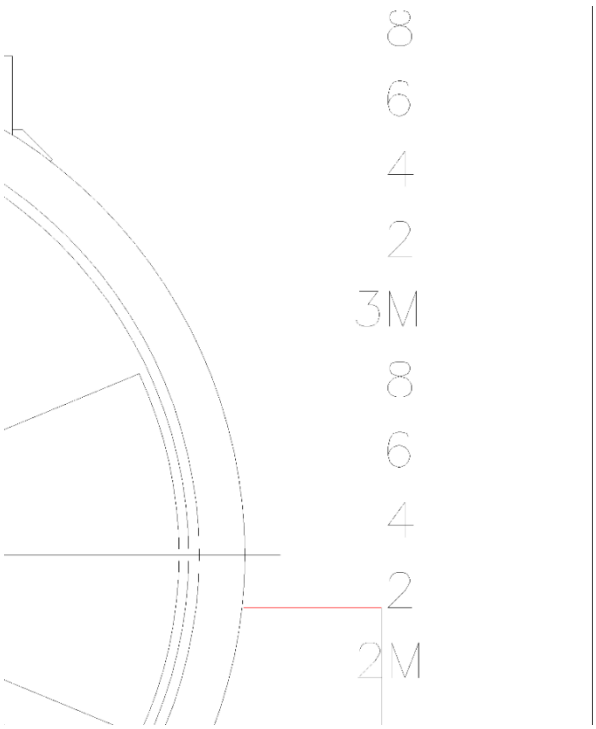


Fig.31: Example of horizontal error detected

ChatGPT is capable of handling multiple images per prompt. With some testing, the limit was found to be three images with maximum resolution. For smaller images, the maximum allowable upload amount could be reached, which is ten per prompt.

Considering time usage is an important aspect of this research, five separate and new analyses were conducted. All used the same ten Vard images and method description to create a structured evaluation procedure, which would collect the most accurate and representative time usage data possible. ChatGPT is not capable of extracting information from or between different conversations when memory is turned off. Though, to ensure new conversations could not get any advantages from previous conversations, the name of each image was given a different number. Analysis A had images 1 to 10, B had 11 to 20, who are the same as 1 to 10, and so on. The running order of each image was also changed as an additional safety measure to impose independence. This setup allowed for a direct comparison of time spent under varying conditions, such as altered image order and parameter values, while keeping the drawing content consistent. Table III provides an overview of the running order of each image and the grouping of some. The names have all been changed back to 1 to 10 for easier comparison. For analysis B and C, the running order is the same, but the parameter values are different.

Table IV presents the parameters used for each analysis, the time measurements collected and other relevant data. Fig.33 compares the time spent on each run for all five analyses.

Table III: Running order of images used in each analysis

	Analysis				
Run number	A	B	C	D	E
1	1	10	10	1, 10	1, 2, 3
2	2	9	9	2, 9	8, 9, 10
3	3	8	8	3, 8	4, 5, 6
4	4	7	7	4, 7	7
5	5	6	6	5, 6	
6	6	5	5		
7	7	4	4		
8	8	3	3		
9	9	2	2		
10	10	1	1		

Table IV: Selected data for each analysis and the different time measurements taken

Analysis	**A**	**B**	**C**	**D**	**E**
(*min_line_length*, *search_radius*)	(300, 5)	(300, 5)	(100, 3)	(300, 5)	(300, 5)
(Number of images per run, number of runs)	(1, 10)	(1, 10)	(1, 10)	(2, 5)	(3, 3) and (1, 1)
Total analysis time	6:44 min	9:33 min	8:48 min	5:22 min	4:51 min
Average analysis time per image	40.4 sec	57.3 sec	52.8 sec	32.2 sec	29.1 sec
Approximate total time writing promts	9 min	8 min	8 min	10 min	2 min
Total time	15:44	17:33	16:48	15:22	6:51

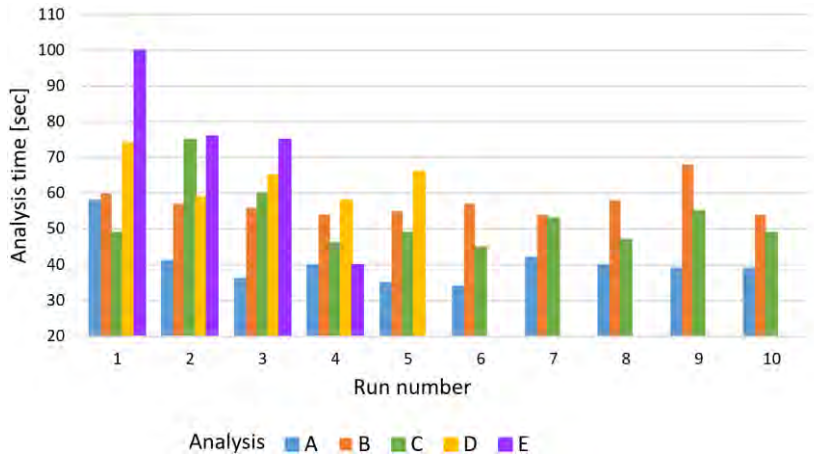


Fig.33: Graph displaying time usage of ChatGPT's Vard analyses

5. Discussion of the Findings

5.1. Method Functionality and Parameter Design

The developed method successfully detects straight-line misalignments by analyzing pixel connectivity in pixel images. Fig.34 for a flow chart guide to the process.

The developed method successfully detects straight line misalignments by using two adjustable parameters, minimum_line_length and search_radius. These settings are important to avoid marking every line and require some understanding of the type of error sizes expected in the drawings. Since

each pixel in the image represents one millimeter on the drawing, a setting of 300 pixels for line length and 5 for search radius means the system targets lines at least 30 cm long that are misaligned within 5 mm of each other.

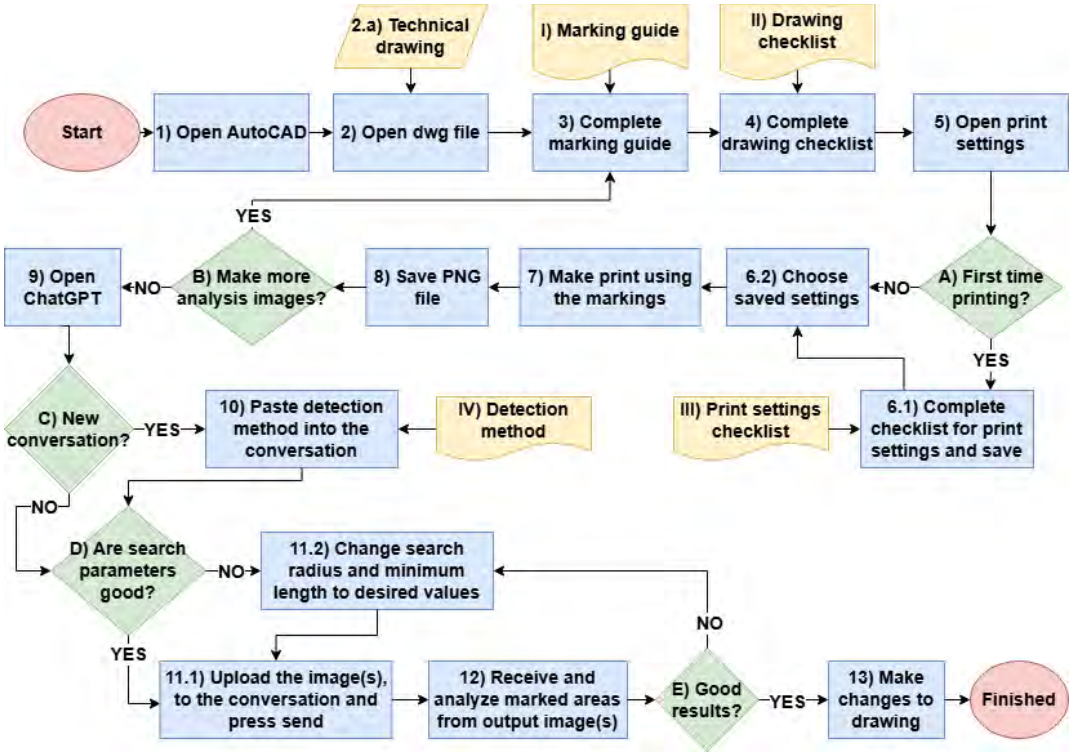


Fig.34: Flow chart of the image creating and analyzing process

5.2. Limitations Due to Image Resolution and Analysis Setup

ChatGPT's resolution limit introduces challenges when analyzing larger drawings. To avoid missing errors in images' border areas, multiple overlapping images must be created. This increases time and labor. While ChatGPT can process up to 10 images per prompt, testing shows that only three high-resolution images can be reliably processed before exceeding the model's limit.

5.3. Time Measurement and Case Study Observations

The Vard general arrangement tests provided valuable data on time usage and performance. While ChatGPT correctly identified all intended misalignments within parameter definitions, the overall analysis time did not offer significant labor savings compared to manual review. The process still requires human attention for image preparation, uploads, and prompt setup.

5.4. Future Expansion and Practical Considerations

Although the method is limited to vertical and horizontal misalignments, more structured error descriptions can be developed. Currently, the method serves best as a digital assistant for quick, internal checks, not as a replacement for classification inspections or full manual review. With further development, the tool could reduce inspection workloads or help identify issues earlier in the design process.

6. Conclusion: Contribution and Future Work

6.1. Summary of the Developed Method

The developed method works for its intended purpose, but requires improvements to become a useful

tool.

This paper presents the creation of a method that uses ChatGPT to visually detect straight line misalignments in black-and-white PNG images of ship drawings. The method relies on two user-controlled parameters, minimum_line_length and search_radius. When tuned appropriately, these parameters allow the AI to highlight both misaligned and closely placed parallel lines, helping engineers identify potential placement errors, although the output images reduce the area that needs to be manually reviewed, the total time spent preparing and uploading images limits immediate time savings. Still, the work demonstrates that even general-purpose large language models like ChatGPT can perform pixel-based visual analysis, without needing custom coding or training.

6.2. Reflections on AI Selection and Testing Process

Large language models and computer vision tools were chosen due to their accessibility and low technical barrier, requiring no programming or data handling skills. Among those tested, ChatGPT proved to be the most knowledgeable and consistent when interpreting ship drawing data. These tests were performed in the spring of 2025, and output can potentially change with updates to the program. Key findings were the use of pixel-based inspections and the usefulness of feedback from ChatGPT to refine the prompts, improving accuracy and reliability. This highlighted the importance of precise communication when working with AI.

6.3. Limitations and Opportunities for Improvements

Currently, the method is limited to detecting misalignments in vertical and horizontal lines only. While it can support inspections, it cannot yet replace them due to the need for manual preparation and supervision. With further development, the method could evolve into a more robust assistant. Future work will follow three main development paths: adding more error type descriptions, automating image generation and printing, and dividing high resolution drawings into sub-images suitable for ChatGPT. These steps will each include performance checks before adding them to the method description. Continued advancements in AI and automation will further increase the method's value and integration potential in real-world ship design tasks.

Acknowledgements

The research is partially supported by the EU Horizon project Smart European Shipbuilding (SEUS) Project (grant ID 101096224). SEUS is dedicated to advancing computational tools to enhance ship design and construction efficiency while also raising awareness of the current state of the art in maritime digitalization (https://seus-project.eu/). Appreciation is also given by VARD Group AS (Ålesund, Norway) for the research idea, support of staff, as well as feedback on case studies and results.

References

DZHUSUPOVA, R.; BANOTRA, R.; BOSCH, J.; OLSSON, H.H. (2023), *Using artificial intelligence to find design errors in the engineering drawings*, Wiley

EIDE.I.S.; WEIDEMANN, M. (2023), *Concept Development of a Sailing Cruise Ship Powered by Alternative Fuel*, bachelor thesis, NTNU, Ålesund

GASPAR, H.M. (2016), EMIS *Project Report*, RFF/NTNU/Ulstein, Ålesund

GASPAR, H.M.: BERTRAM, V. (2023), *The Use of ChatGPT and Similar AI in Marine Engineering: Limitations and Opportunities*, 22nd COMPIT Conf., Drübeck

GASPAR, H.M.; BERTRAM, V.; PLOWMAN, T. (2023), *Hic Rhodus, Hic Salta: ChatGPT and other A.I. Tools for Maritime Applications*, 15th HIPER Symp., Bernried

WEIDEMANN, M. (2025), *AI Assisted Misalignment Detection in Technical Ship Drawings*, NTNU Ålesund

Interactive Reduced Order Models for Ship Hull Design and Optimization Coupling an Open-Source CFD Tool with Advanced RBF Mesh Morphing

Emiliano Costa, Benedetto Di Paolo, Apostolos Krassas, Paolo Geremia, ENGYS, Trieste/Italy, e.costa@engys.com
Corrado Groth, Marco Evangelos Biancolini, University of Rome "Tor Vergata", Rome/Italy
Marco Camponeschi, Ubaldo Cella, Emanuele Di Meo, RBF Morph, Monte Compatri/Italy

Abstract

We present the development of the proof-of-concept (PoC) for an effective and efficient numerical procedure conceived to design and optimize ship hulls by leveraging reduced order models (ROMs) gained processing a database of computational fluid dynamics (CFD) fields relevant to the marine sector that is created by performing a design of experiments (DOE) study parametrizing the CFD case using radial basis functions (RBF) mesh morphing. The ROMs, that can be interactively used to get real-time calculations, demonstrated a maximum percentage error on prediction of less than 3.5% for resistance estimates at design points outside the original DOE set, indicating strong predictive accuracy. The resulting ROMs serve as a foundational step toward building Digital Twins of ships.

1. Introduction

The optimization of ship hull forms has traditionally relied on high-fidelity Computational Fluid Dynamics (CFD) simulations to accurately capture the complex hydrodynamic phenomena governing resistance, propulsion, and seakeeping performance. While CFD provides a robust and reliable framework for performance assessment, its application in design optimization is often hindered by the high computational cost associated with repeated simulations across a large design space. This limitation becomes particularly pronounced when gradient-based or evolutionary optimization strategies are employed, as these approaches may require thousands of design evaluations.

Recent advances in Artificial Intelligence (AI) and Reduced-Order Modeling (ROM) techniques have introduced promising alternatives to accelerate hull-form optimization without compromising predictive accuracy. ROMs, constructed from a limited set of high-fidelity CFD solutions, enable the rapid evaluation of new design candidates at a fraction of the computational cost. These surrogate models, which may be based on methods such as Proper Orthogonal Decomposition (POD), Dynamic Mode Decomposition, or Machine Learning regression techniques, capture the dominant flow features while discarding less influential dynamics. When integrated into optimization frameworks, ROMs allow for efficient exploration of the design space, enabling near real-time prediction of hydrodynamic responses.

In the context of simulation-based design optimization in marine engineering, *Serani et al. (2024)*, we performed a proof-of-concept (PoC) to showcase the streamlined and effective numerical procedure conceived to design and optimize ship hulls by leveraging ROMs gained by treating a dataset of results assessing CFD fields relevant to the marine sector. Such dataset is created by carrying out a Design of Experiments (DOE) study parametrizing the CFD case using RBF mesh morphing.

2. Rationale of the numerical procedure

The proposed numerical procedure aims at designing and optimizing ship hull forms by leveraging ROMs obtained processing a database of CFD fields relevant to the marine sector, which is generated by carrying out a DOE study parametrizing the shape of the CFD case using Radial Basis Functions (RBF) mesh morphing techniques, *Biancolini (2017),* and interactive reduced order models, *Biancolini (2020)*.

The whole numerical procedure consists of two sequential stages: an off-line stage and on-line stage. To perform the tasks of such stages, different CAE software tools were employed, while Python scripts were suitably implemented to streamline the processes. The key aspects and steps of those stages are respectively described in the following sections.

2.1. Off-line stage

The off-line stage consists of the following steps:

- Definition of the CFD baseline case: using an open-source CFD software, the reference (i.e., baseline) case is suitably defined to simulate and create the sought numerical output of interest of the PoC.
- Parametrization of the reference CAD model of the hull: the selected CAD model is geometrically parametrized using RBF mesh morphing techniques thus creating the shape variants of the baseline configuration of the ship's hull.
- Generation of the CFD database: the database of the CFD results of interest (e.g., resistance, wall shear stress, free surface elevation) is created performing a DOE study, where each design point (DP) consists of a CFD simulation of calm-water resistance of a CFD case variant, adopting the Linearized Free Surface (LFS) (*Geremia 2019)* solver of the open-source CFD tool used.
- ROMs generation: a ROM is created for all CFD fields of interest such as mesh, resistance and pressure over the hull surface, by processing the CFD results database using the Proper Orthogonal Decomposition (POD) method, *Ostrowski et al. (2008), Dotta et al. (2021),* based on truncated Singular Value Decomposition (SVD), *Stewart (1993).*

The workflow of the off-line stage is shown in Fig.1.

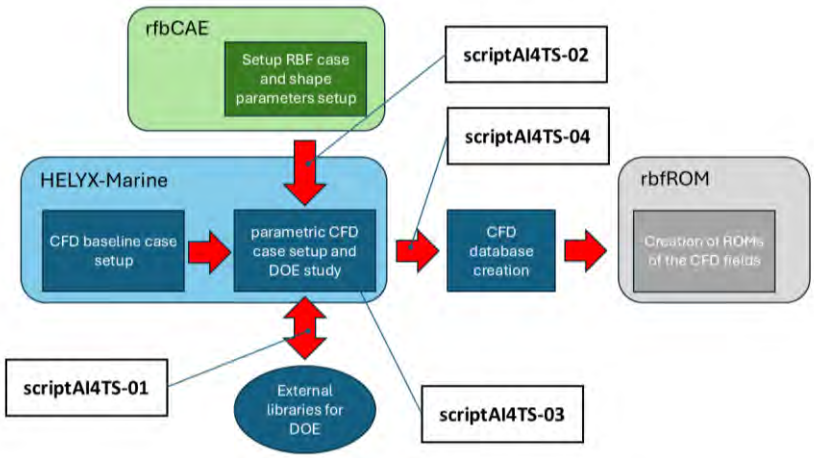


Fig.1: Workflow of the off-line stage

2.2. On-line stage

The on-line stage consists of the following steps:

- Real-time prediction using ROMs: ROMs can accurately predict both the ship resistance and distribution of certain relevant CFD quantities both using text commands and through an interactive dashboard developed using PyVista, https://docs.pyvista.org/, enabling real-time 3D visualization of the CFD results and quantities of interest for new unseen geometries defined with new combinations of parameters not belonging to the training set.
- ROM post validation: identified candidates undergo a full validation by full high fidelity CFD. Thanks to the high level of automation this task can be enabled on-line.

The workflow of the on-line state is shown in Fig.2.

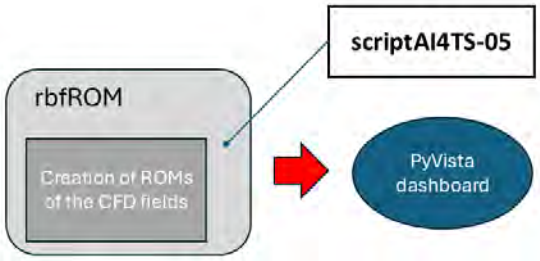


Fig.2: Workflow of the on-line stage

2.3. Automation

In order to make the proposed numerical procedure more efficient while limiting as much as possible the intervention requested to the user, Python scripts were developed and used. This programming language was deliberately chosen, as it enables the automation of CFD and mesh morphing operations through dedicated Application Programming Interfaces (API).

The operations that are automated by these scripts are shown in Fig.1 and Fig.2, respectively. Several stand-alone scripts were implemented to keep the approach more general and flexible; the scripts can be merged or customized depending on specific needs (e.g., in the case that CFD calculations are requested to be accomplished on High-Performance Computing infrastructures).

3. Baseline CFD case generation and results

3.1. CFD methodology

In the field of ship hull hydrodynamics, CFD solvers are commonly built upon either the Volume-of-Fluid (VOF), *Hirt and Nichols (1981),* or level-set approaches, *Yang et al. (2007)*. These techniques are typically integrated within the Reynolds-Averaged Navier-Stokes (RANS) equations using a finite-volume discretization. Both methods have shown excellent accuracy in estimating hull resistance and other critical hydrodynamic characteristics. However, their reliance on high-performance computing and extended runtimes makes them less suitable for use during the early design phases, where rapid feedback is essential.

Additionally, the stability and precision of VOF and level-set solvers depend heavily on tailored mesh generation strategies and the use of very small time steps. These requirements add complexity and further increase computational demands. To mitigate these challenges, a RANS-based Linearized Free surface (LFS) solver that incorporates viscous effects was developed. This solver operates under a steady-state assumption, enabling significantly faster evaluations of hull resistance and related parameters, while maintaining a high level of accuracy, *Rosemurgy et al. (2011)*.

The LFS solver is derived from the linearized form of the unsteady Neumann-Kelvin boundary-value problem, which models ship-generated waves. It assumes that wave amplitudes and slopes are small enough to justify linearizing the free surface boundary condition on the undisturbed water surface. This simplification allows for a double-body representation and the use of a single-phase flow solver, making it possible to analyze multiple hull motions efficiently and with reduced computational effort. Validation of the RANS solver was conducted using the Gothenburg 2010 Case 2.2b, which involves the Kriso Container Ship (KCS) at model scale, http://www.simman2008.dk/KCS/kcs_geometry.htm. Results were compared against a high-fidelity VOF simulation and experimental data. The computational mesh consisted of 1.2 million cells for the LFS case and 3.1 million cells for the VOF case. Fig.3 shows a close-up of the mesh configurations.

The free surface elevation results are presented in Fig.4. The resistance coefficient is shown in Fig.5:

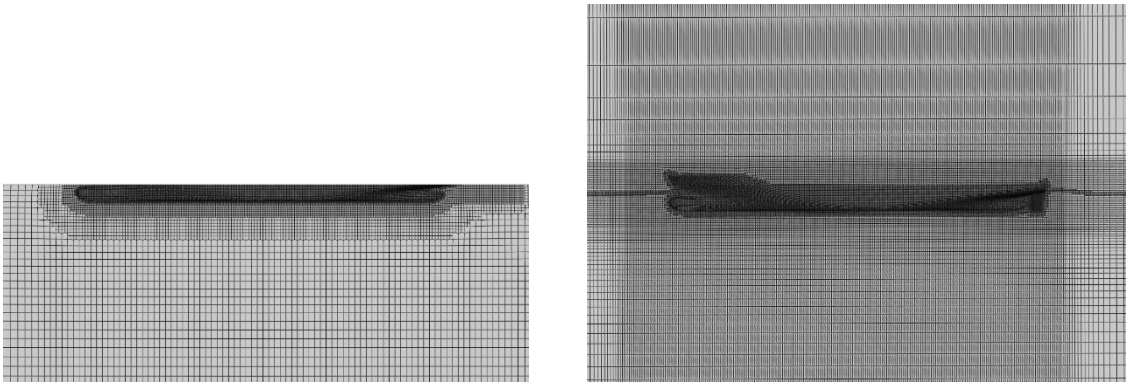


Fig.3: Computational mesh for the validation case: LFS solver (left) and VOF case (right)

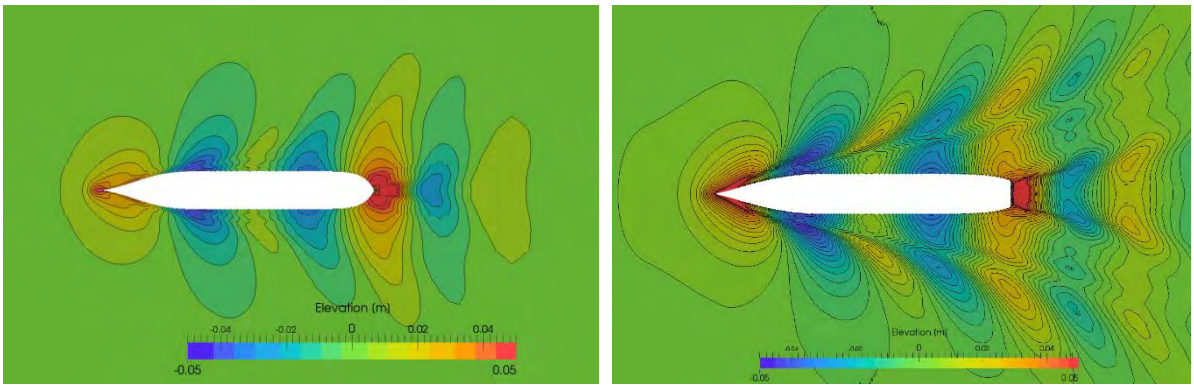


Fig.4: Free surface elevation results for the validation case: LFS solver (left) and VOF case (right)

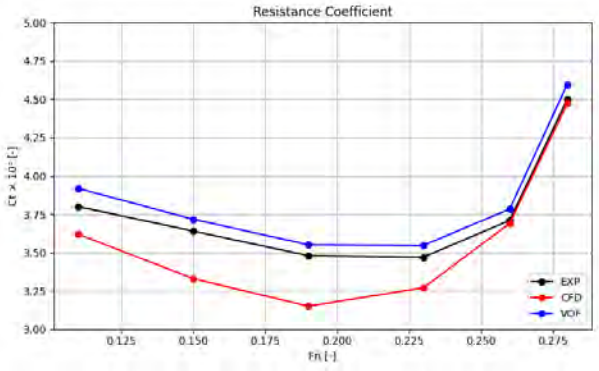


Fig.5: Resistance coefficient for the validation case

Although the LFS method is an approximation, the resistance force trend is accurately captured across the full range of Froude numbers. The key advantage of the LFS solver lies in its computational efficiency: each simulation runs in approximately 20 minutes on a 64-core AMD Genoa system, compared to 5.6 hours required by the VOF solver. This makes the LFS approach particularly attractive for parametric studies and hull-form optimization during the early design phase.

3.2. Strategy to set up the RBF case

The baseline CFD case of the KCS container ship was created in the CFD software to calculate the results of interest, which are required for the subsequent development of the ROMs. This process was carried out semi-automatically through the combined use of Python and the CFD software's API.

The procedure involves loading the discretized CAD model of the KCS in STL format, which represents half of the ship model at full scale, and generating the mesh which is properly detailed to

capture the areas of the surfaces with higher curvature. The generated volume mesh consists of approximately 2.5 million cells and includes multiple wall-cell layers to enable the correct application of wall functions in the vicinity of the hull surfaces. Standard water properties were assumed, and the k–ω SST turbulence model was adopted.

To compute the quantities of interest in steady state conditions corresponding to a Froude number (Fn) of 0.26, a RANS simulation was carried out with 5,000 iterations, sufficient to achieve convergence of the calm-water resistance with two degrees of freedom, namely heave and pitch angle, using the LFS solver available in the CFD tool.

Once the steady-state solution was obtained, the CFD results of interest were extracted, namely the total resistance of the hull, the pressure distribution on the hull surface, and the wave elevation field, saved in VTK (*Schroeder et al. 2006)* ASCII format. The evolution of the total force acting on the KCS hull as a function of the CFD iterations is shown in Fig.5. The steady state resistance, calculated as the average of the total force over the final 1,000 iterations, was found to be 102.45 kN. This quantity is the main CFD scalar field of interest considered in the development of the PoC.

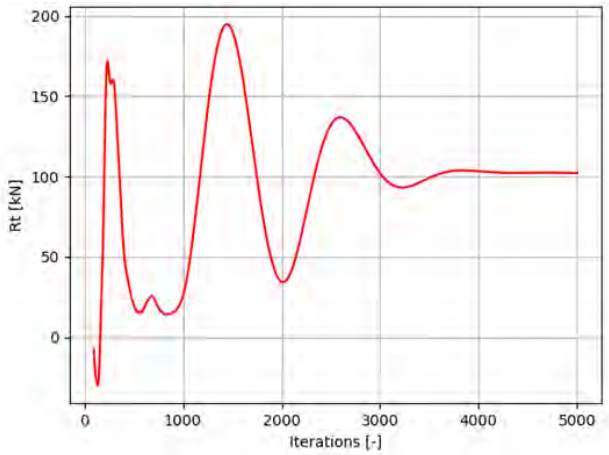


Fig.5: Total force profile as a function of CFD iterations

Fig.6 shows the two main three-dimensional outputs of interest in the PoC, i.e. the steady state pressure distribution over the KCS hull and free surface elevation on the left and right, respectively.

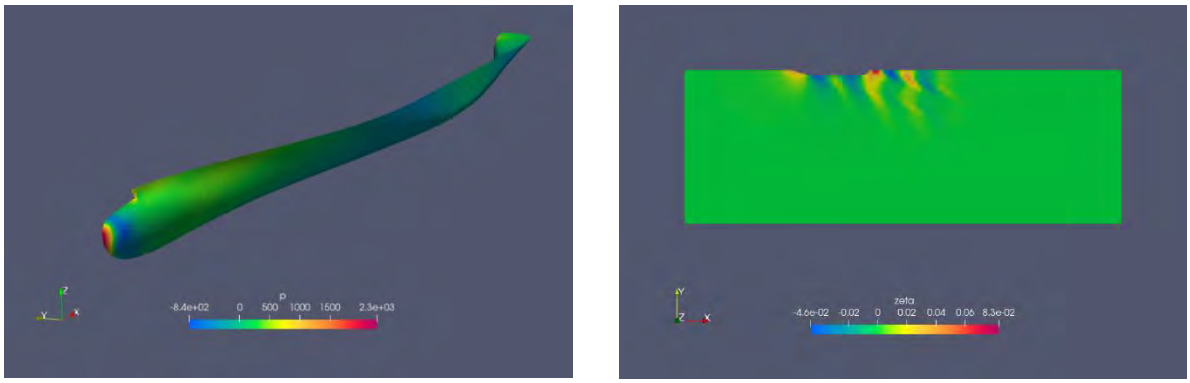


Fig.6: Distribution of pressure over the hull and free surface elevation

Given the need to build a database in the subsequent steps of the numerical procedure, the relevant CFD outputs for the PoC - such as pressure distribution over the hull, hull's resistance and free surface elevation - are configured to be exported in VTK ASCII format upon completion of the computation.

4. Geometrical parametrization using RBF mesh morphing and design points generation

4.1. Strategy to set up the RBF case

In the context of the tools used in the PoC, two morphing strategies can be considered for generating the DOE DPs, namely surface-morphing and volume-morphing.

In the surface-morphing approach, new hull geometries are obtained by applying RBF modifications to a discretized STL reference model. Each variant is then used to create a separate case in the CFD tool. This method is generally more robust, as it modifies only the surface mesh, with no direct impact on the quality of the volume mesh which will have, in general, a different topology for the different DPs.

In contrast, the volume-morphing approach directly alters the baseline CFD mesh by applying combinations of RBF parameters. While this method avoids regenerating a new volume mesh for each DOE DP and can therefore be faster, it is less robust because mesh quality deteriorates as deformations increase, limiting the allowable extension of modifications.

Considering the need for POD mathematical framework to process arrays of CFD fields of consistent length, volume-morphing was employed to ensure the mesh topology remains consistent across the different DPs of the DOE. This guarantees that the CFD results contain the same number of surface data points.

When the validity of volume mesh cannot be guaranteed, a hybrid approach can be used (not covered in this study): both methods are applied simultaneously, so that the new geometry is represented both as a deformed mesh and as a newly generated highly quality mesh. Once CFD convergence is achieved, the results are mapped onto the deformed mesh, which is then ready for POD compression.

4.2. RBF case setup

The adopted strategy for implementing RBF-based shape modifications focused on altering both the bulbous bow and the stern region of the KCS hull, as these areas were identified as having the most significant impact on hydrodynamic performance. The modifications included shape variations both within the hull's symmetry plane and in directions outside of it (i.e., out-of-plane).

To ensure a high degree of flexibility in creating such modifications by means of the RBF mesh morphing tool, the following items were generated and provided as input: (i) a discretized CAD model of the entire ship in STL format, serving as one of the primary inputs for generating the baseline mesh; (ii) a neutral CAD model of the bulbous bow and stern region in STEP format, used as input to the mesh-morphing software for creating the cases defining the RBF shape modifications; and (iii) a discretized CAD model of the bulbous bow and stern region in STL format, used as input to the morphing tool for the same purpose as the neutral CAD model. In such a way both CAD entities such as vertex, edges and areas, as well as mesh entities such as nodes can be used to create RBF shape modifiers.

The discretized CAD model of the entire hull features a high spatial resolution, allowing for an accurate representation of surfaces with significant curvature. In contrast, the discretized CAD models of the bulbous bow and stern region were generated with a lower resolution, to simplify the setup and facilitate the application of shape modifications on the baseline CFD mesh.

The geometric parameterization of the bulbous bow and stern region of the KCS hull was investigated. The mesh-morphing case setup was created using the GUI provided with the morphing software. The CAD models of the bulbous bow and stern region are shown in Fig.7 and Fig.8 respectively, where the neutral CAD model in STEP format is presented on the left, and the corresponding discretized STL model is presented on the right.

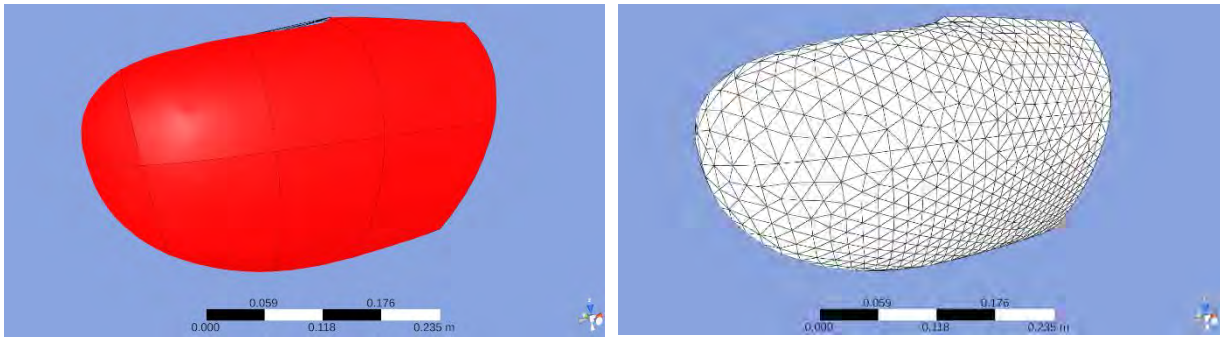


Fig.7: Neutral and discretized CAD models of the bulbous bow

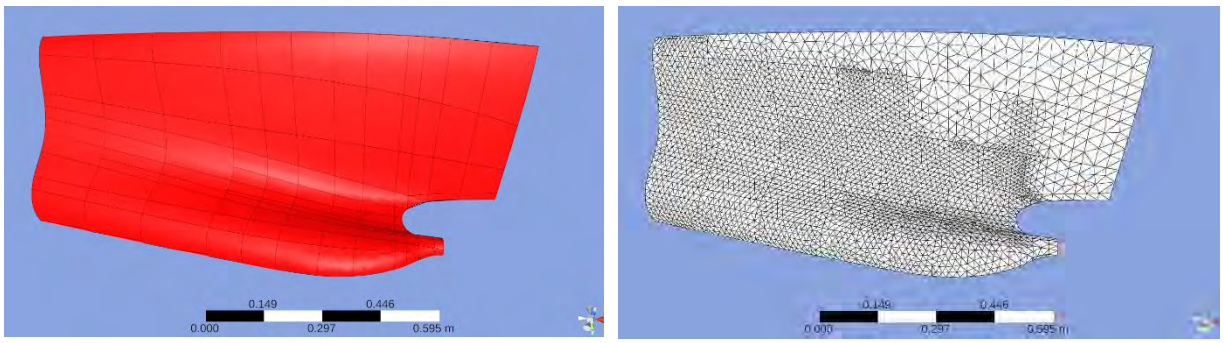


Fig.8: Neutral and discretized CAD models of the stern region of the hull

Utilizing the CAD models mentioned above, a morphing-tool case was generated that includes five different RBF shape modifications: three applied to the bulbous bow and two to the stern region. To ensure continuity of the modifications with respect to the hull, all RBF shape parameters of the first three modifications were constrained to preserve the position of the edges shared with the hull by imposing a zero-displacement condition.

For one of the RBF shape modifiers that acts outside the "symmetry" plane of the model in Fig.9 (left) the positions of the RBF points before and after the application of morphing are shown in green and blue respectively. This modification combines RBF points that keep fixed—including the end nodes of the node set highlighted on Fig.9 (right) —with the scaling action illustrated on the left for a given amplification factor. The scaling operation thus makes it possible to widen or reduce the bulb's cross-sectional area while preserving, as much as possible, curvature continuity in the transverse direction relative to the ship's forward motion.

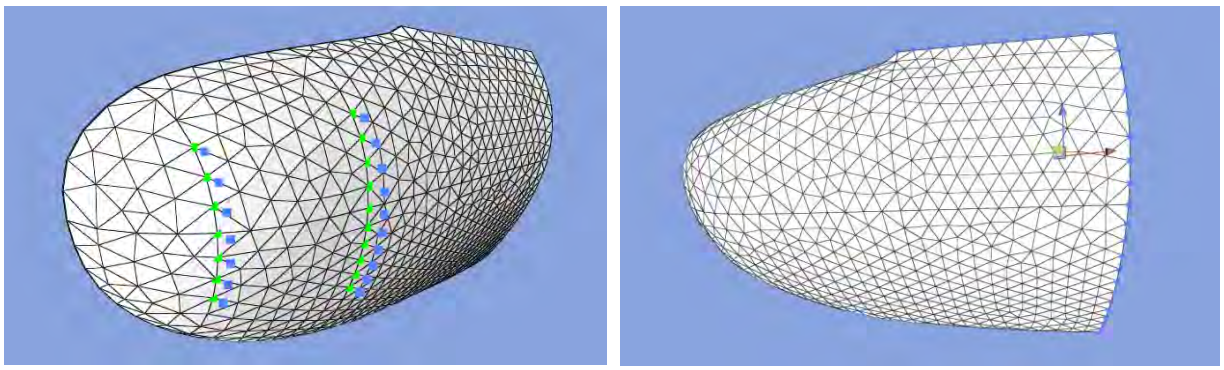


Fig.9: Main settings of one of the out-of-plane shape modifications

Fig.10 shows the effect of the shape modification (right) compared to the baseline bulb configuration (left), from two different views. Once the CFD baseline case and RBF solution to parametrize the hull are prepared, the main inputs needed by the numerical procedure are ready to be used to run the semi-automated processes described below.

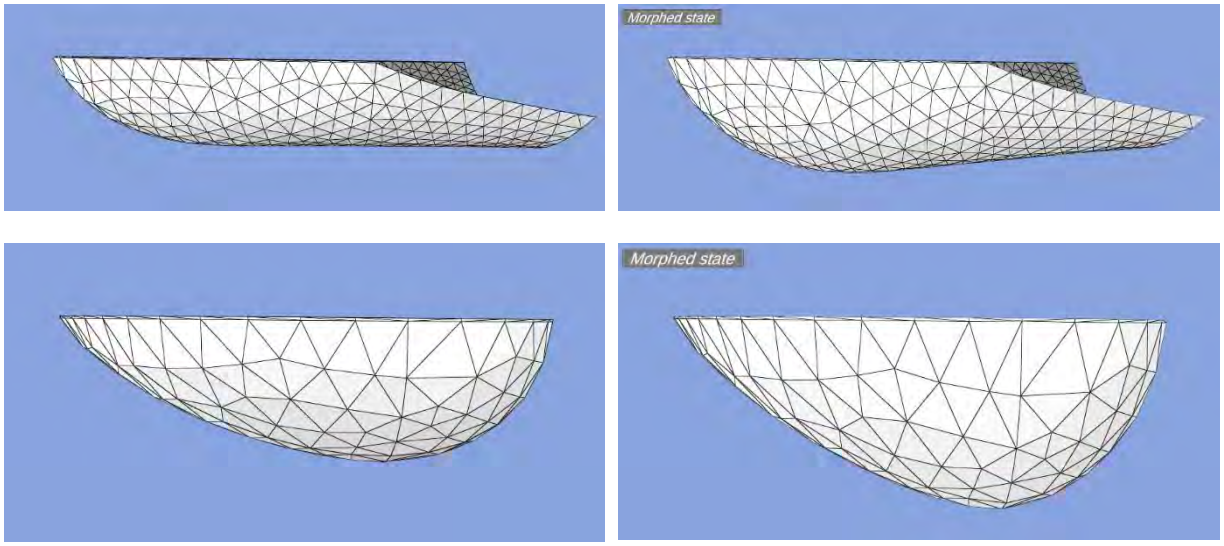


Fig.10: Morphing action of one of the out-of-plane shape modifications

5. CFD database generation and creation of ROMs of CFD fields

5.1. Database generation

A first Python script, Fig.1, automatically generates the DOE table in a format suitable for the mesh morphing tool by reading the main data defining the number of RBF parameters, their name and range of variation, as well as the number of DPs and the sampling method in an input file. The DOE study carried out for the PoC was generated via Latin Hypercube Sampling (LHS) and finally composed by 101 DPs including the baseline configuration.

The second Python script automatically creates the DPs as CFD morphed cases configurations, by reading the DOE table generated in the previous step adopting the volume morphing approach.

5.2. Database generation

The third Python script manages the simulation of all DPs that are run in sequence. The values of the hull's resistance calculated for all DPs vary between -1.43% and +10.14% with respect to the one determined for the baseline configuration of the KCS hull. The machine to perform this task was equipped with 2x EPYC 9354 (Genoa) 32-Core with a clock frequency of 3.5 GHz. After the CFD computation, the fourth and final Python script collects all the CFD results of interest and generates the CFD database. This includes a VTK file containing the pressure distribution over the hull and free surface elevation, as well as the DOE table enriched with resistance values for each DP.

5.3. ROMs creation

The first task for ROMs creations consists of the setup of the framework suitable for POD processing according to which the CFD fields of interest, that is mesh, resistance, pressure and free surface elevation, are structured as arrays.

Once this framework is established, the weights and parameters of the CFD fields are determined. Subsequently, the POD processing is performed to compute the modes of the CFD fields. ROMs can then be constructed by retaining a selected number of modes—an operation commonly referred to as modes truncation. This number of modes can be chosen by the user based on graphs showing the maximum error associated with the model basis, which is computed during the POD processing. An example of these graphs is shown in Fig.11 referring to both the mesh and pressure field.

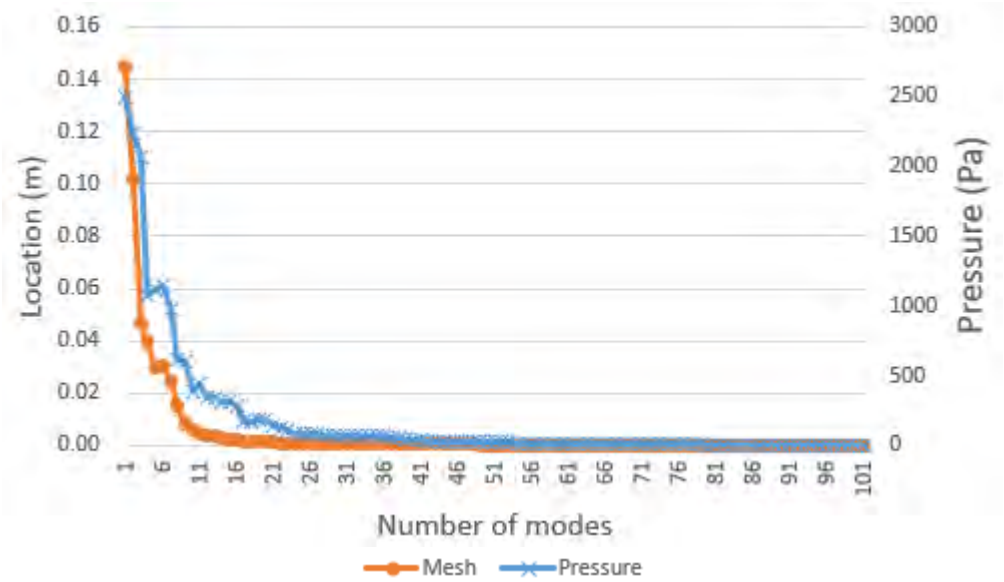


Fig.11: Maximum error of mesh and pressure against number of modes

Considering the low computing demands related to the update of the CFD fields through ROMs, all the computed modes were used and the full SVD decomposition retained.

6. Results

To assess the accuracy of the ROMs during the on-line phase, the evaluation first focused on the prediction of pressure and resistance fields. These verifications were performed for two design points within the DOE as well as for two additional points outside the DOE table using both text commands and the interactive dashboard.

6.1. Verification using DOE design points and additional design points

Table I reports the results of the resistance prediction accuracy verification for the selected DPs, showing that the error is negligible; this is a direct consequence of having used all the modes as the SVD decomposition is complete and no truncation has been imposed.

Table I: Validation of ROM accuracy for resistance at the four DOE DPs

ID	RBF parameters combination	CFD	ROM	Error %
0	0, 0, 1, 0, 1	102.4538300	102.45383	0.000000E+00
21	-0.0347, 0.0327, 1.19, 1.6, 1.25	108.5159485	108.51595	-1.382285E-06

Fig.12 and Fig.13 present, respectively, a visualization of the pressure distribution and of the free surface elevation predicted by the interactive ROM and computed with the full CFD. As shown, the dashboard consists of a panel displaying the mesh, the scalar field value of hull resistance in the top-right corner, and the distribution of three-dimensional CFD outputs such as pressure and free surface elevation. Additionally, a set of five sliders on the left allows users to adjust the amplitude of each RBF parameter within its variation range to update the CFD fields accordingly.

To complete the verification of the ROM prediction accuracy, two DPs outside the DOE set were identified. The combinations of the RBF parameter values defining these points are collected in Table II. The maximum error is quantified as the capability to predict the resistance variation; it is equal to 3.5% for the additional study point with ID1, and equal to 0.03% for the additional study point with ID2.

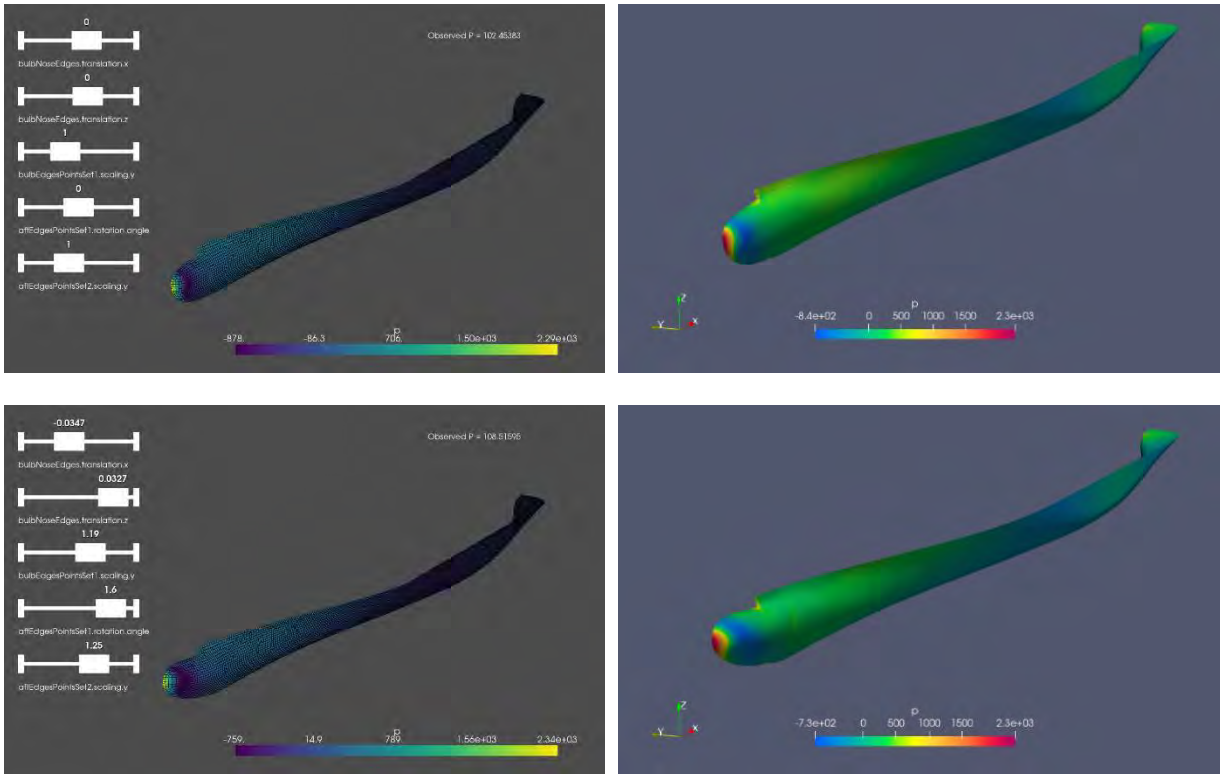


Fig.12: Validation of ROM accuracy for resistance and pressure at two DOE DPs, ROM (left) and CFD (right)

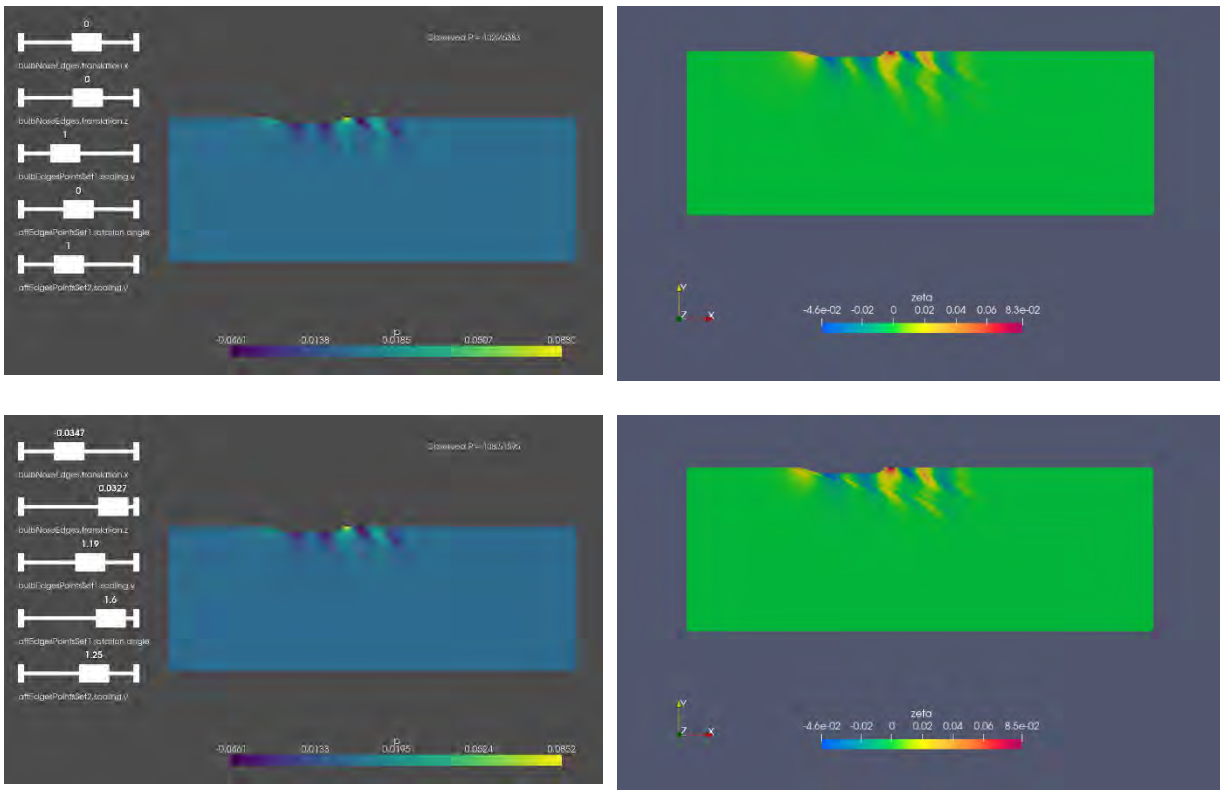


Fig.13: ROM accuracy verification for resistance and free surface elevation for two DOE DPs, ROM (left) and CFD (right)

Table II: Validation of ROM accuracy for resistance at the two additional DPs

ID	RBF parameters combination	deltaCFD	deltaROM	Error %
1	-0.05, 0.0392, 0.9, 1, 0.9	2.58	2.49	3.48
2	0.0209, -0.0312, 1.04, 1.82, 1.16	1.14	1.14	0.03

Fig.14 shows the pressure distribution on the hull and the resistance values for the selected out-of-DOE points respectively predicted by ROMs compared with the full CFD.

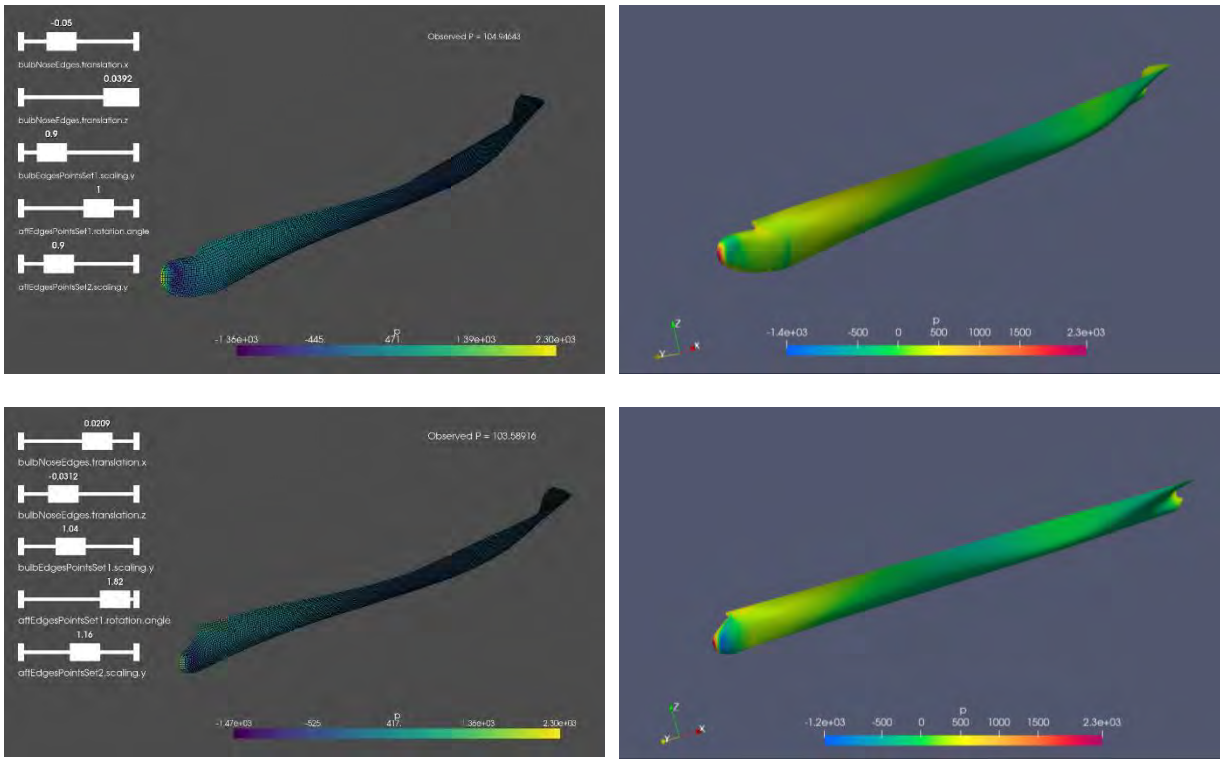


Fig.14: Validation of ROM accuracy for two DPs outside of the DOE

6.2. Real-time computations and interaction with the dashboard design points

The final evaluation of the numerical procedure is focused on the user experience during the on-line stage, specifically in terms of interactivity and response times when visualizing the CFD fields provided by the ROMs through the dashboard, as a function of different combinations of RBF shape parameters. Overall, the user experience proved to be highly satisfactory, as demonstrated by the animations available online on a proprietary YouTube channel, Interactive ROM predictions of pressure over the KCS hull and KCS hull resistance, Interactive ROM predictions of free surface elevation and KCS hull resistance.

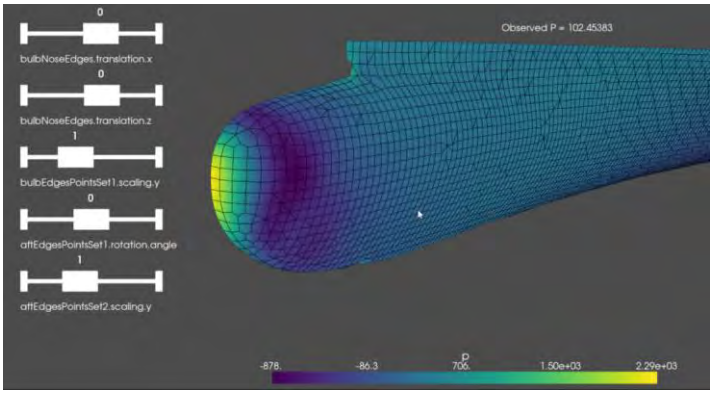


Fig.15: Snapshot of on-line stage to compute hull resistance and pressure distribution interactively

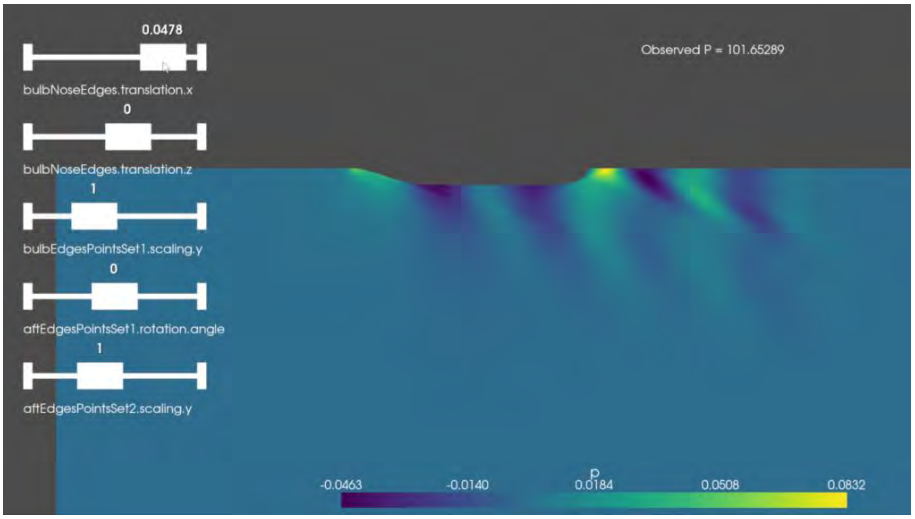


Fig.16: Snapshot of on-line stage to compute hull resistance and free surface elevation interactively

Two snapshots captured from these animations are shown in Fig.15 and Fig.16 for the pressure distribution over the KCS hull and free surface elevation, respectively. For both hull resistance of the current RBF parameters combination is reported as well.

The assessment of interactivity with the ROMs was carried out using a laptop computer equipped with an AMD Ryzen 7 PRO 6850U processor operating at a clock frequency of 2.70 GHz and integrated with an AMD Radeon 680M graphics card.

7. Conclusions

In this paper, the development of a Proof-of-Concept demonstrating a streamlined numerical procedure for ship hull design and optimization was presented. The procedure leverages Reduced-Order Models (ROMs) constructed through the Proper Orthogonal Decomposition (POD) method, applied to a dataset of Computational Fluid Dynamics (CFD) field data relevant to the marine sector. The database was generated by conducting a Design Of Experiments (DOE) study, consisting of 101 design points obtained by parameterizing the CFD model of the KCS hull with five shape parameters, implemented via Radial Basis Function (RBF) mesh morphing. Each design point was simulated using a steady-state RANS solver within an open-source CFD tool to compute calm-water resistance, thereby assembling the full database of morphed hull shapes.

The ROMs enabled real-time predictions, accessible through either text commands or an interactive dashboard that can be run even on a personal laptop smoothly, with high accuracy at design points outside the DOE set, the maximum relative error in resistance prediction was below 3.5%. Wherever possible, the workflow was automated through Python scripting, further enhancing the efficiency and reproducibility of the procedure.

RBF mesh morphing proved to be a versatile tool for handling complex shape parameterization, offering a high level of control over surface modifications, while POD processing demonstrated to be fast, robust, and computationally efficient. The resulting ROMs open the way for multi-physics studies such as fluid–structure interaction (FSI) and represent a key step toward the development of Digital Twins of ships. In this context, the adoption of open-source CFD tools for large-scale computations makes the overall approach economically attractive, as no additional licensing costs are incurred. This aspect is particularly relevant when exploiting HPC resources to generate large datasets for data-driven modeling.

Acknowledgements

This work was developed within the framework of the Italian research project AI4TwinShip, that was co-funded by the European Union, the Italian government and the Friuli-Venezia Giulia region in the context of the PR FESR 2021-2027 program.

References

BIANCOLINI, M.E. (2017), *Fast Radial Basis Functions for Engineering Applications*, Springer Cham, https://doi.org/10.1007/978-3-319-75011-8

BIANCOLINI, M.E.; CAPELLINI, K.; COSTA, E.; GROTH, C.; CELI, S. (2020), *Fast interactive CFD evaluation of hemodynamics assisted by RBF mesh morphing and reduced order models: the case of aTAA modelling*, Int. J. Interact. Des. Manuf. 14, pp.1227–1238

DUTTA, S.; FARTHING, M. W.; PERRACCHIONE, E.; SAVANT, G.; PUTTI, M. (2021), *A greedy non-intrusive reduced order model for shallow water equations*, J. Computational Physics 439, 110378

GEREMIA, P.; MAKI, K.J.; ALEXIAS, P. (2019), *A linearized free surface RANS method for self-propulsion and maneuvering*, 11th Int. Workshop on Ship and Marine Hydrodynamics (IWSH2019)

HIRT, W.C.; NICHOLS B.D. (1981), *Volume of fluid (VOF) method for the dynamics of free boundaries*, J. Computational Physics 39, pp.201-225

OSTROWSKI Z., BIAŁECKI R.A.; KASSAB A.J. (2008), *Solving inverse heat conduction problems using trained POD-RBF network inverse method*, Inverse Problems in Science and Engineering 16:1, pp.39-54

ROSEMURGY, W.J.; EDMUND D.O.; MAKI K.J.; BECK R.F. (2011), *A Method for Resistance Prediction in the Design Environment*, 11th Int. Conf. Fast Sea Transportation

SCHROEDER, W; MARTIN, K.; LORENSEN, B. (2006), *The Visualization Toolkit* (4th ed.), Kitware

SERANI, A.; SCHOLCZ, T.P.; VANZI, V. (2024), *A Scoping Review on Simulation-Based Design Optimization in Marine Engineering: Trends, Best Practices, and Gaps*, Arch. Computat. Methods Eng. 31, pp.4709–4737

STEWART, G.W. (1993), *On the early history of the singular value decomposition*, SIAM Review 35 (4), pp.551–566

YANG, J.; SAKAMOTO, N.; WANG, Z.Y.; CARRICA, P.; STERN, F. (2007), *Two Phase Level-Set/Immersed-Boundary Cartesian Grid Method for Ship Hydrodynamics*, 9th Int. Conf. Numerical Ship Hydrodynamics, Ann Arbor

Anti-Roll Apps for Container Ships

Nils Otten, DNV, Hamburg/Germany, nils.otten@dnv.com
Ole Hympendahl, DNV, Hamburg/Germany, ole.hympendahl@dnv.com
Karsten Hochkirch, DNV, Hamburg/Germany, karsten.hochkirch@dnv.com

Abstract

This paper describes DNV's anti-roll apps Anti-Roll Assist and Anti-Roll Alert that were developed as smart decision aids for container ships. The apps are based on meta-modelling ship specific extensive numerical simulations of synchronous and parametric rolling. Anti-Roll Assist requires either manual data entry or integration with third-party software such as StormGeo's s-Planner. Anti-Roll Alert has minimized manual input and provides a largely automated system.

1. Introduction

Some 1300 containers per year are lost at sea. Container loss at sea is still relatively rare, considering that approximately 250 million containers are shipped across the world's oceans every year. Yet those few incidents make big waves in the media and can damage the reputation of the ship owner and operator, in addition to the financial loss. Furthermore, lost containers floating in the water, Fig.1, are a hazard for ocean traffic, especially for smaller ships.

Fig.1: Lost containers floating at surface posing threat to shipping, source: Dutch Coastguard

Many of the container losses are suspected to be caused by human errors, particularly oversights in proper lashing. However, excessive roll with associated excessive roll acceleration and forces is also playing a significant role in container losses.

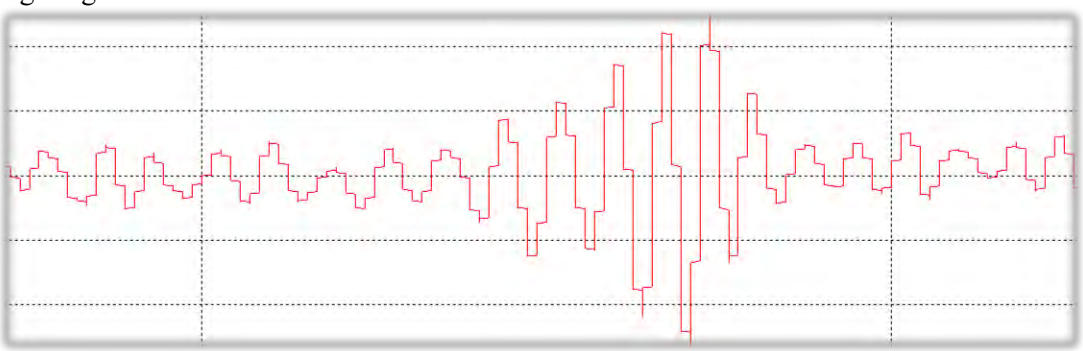


Fig.2: Roll motion in a parametric roll event (roll period approx. 20 s)

In synchronous or parametric rolling, see Appendix, extreme roll angles can happen very suddenly and are difficult to anticipate. Fig.2 shows roll angles for a ship experiencing parametric rolling. After a period of normal, small-amplitude roll, the ship is suddenly excited in parametric roll resonance, and the roll amplitude triples within just a few periods. There is too little time for the crew to take any countermeasures.

It is not really feasible to design or modify container ships, so that they are not susceptible to extreme roll, except using rather expensive active roll damping devices. But there are economical solutions for smart decision support with little interference to the general operation. The DNV approach is to focus on the specific roll problem and to control the risk by giving operational advice. Corresponding apps have been developed in the past few years, and they are described in the following.

2. Anti-Roll Apps

The apps are fast decision support systems for ship crews. In order to give decision support for such complex phenomena like parametric rolling, the apps need to have "intelligence inside", i.e. a knowledge base on the nonlinear motion behaviour of the ship in waves, or in modern parlance a "digital twin". The building of this knowledge base requires extensive preprocessing with nonlinear seakeeping simulations, varying various key variables describing load conditions and seaway.

The ship specific results of the nonlinear seakeeping analysis are saved in a response database. This database is connected to our central server process, which can run locally on a ship or in the cloud. The Anti-Roll Assist app communicates with this server process and sends the input data, such as the operating condition and the sea state parameters. In return it receives the "load on the lashing system". This value is displayed in the app as quasi-instantaneous decision support in the form of polar diagrams with an intuitive colour scheme (blue = low load, red = high load).

Fig.3 illustrates the process of response database creation and the subsequent rapid response app, which will be explained in more detail in the following subchapters.

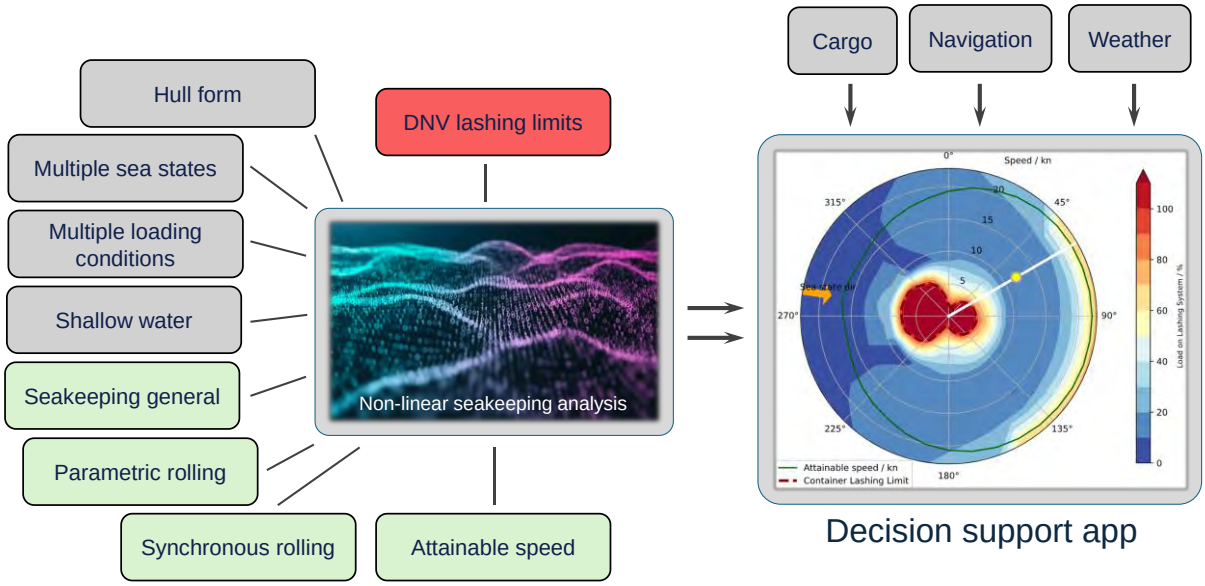


Fig.3: Knowledge base creation (left) and decision support app (right)

2.1. Preprocessing simulations

The decision support apps require an extensive hydrodynamic knowledge base to assess risks of rolling and associated loads on lashing systems. This knowledge base and the subsequent apps are ship specific.

The preprocessing involves comprehensive seakeeping analyses with the actual ship geometry, covering a wide variation of ship operational parameters (mean draft, trim, metacentric height, heading, speed), seaway parameters (significant wave height and peak or mean frequency for wind waves and swell), and water depth.

For the simulations, two DNV in-house ship motion solvers are applied:

- GL-rolls, *Söding et al. (2013)*, uses linear hydrodynamics for those degrees of freedom, for which nonlinearities are insignificant (sway, heave, pitch and yaw), and nonlinear models for the strongly nonlinear degrees of freedom (roll and surge). It simulates the vessel motions in the time domain, which is a pre-requisite to detect parametric rolling. This code is used for all simulations in deep water.
- GL-Rankine, *Söding et al. (2012), Shigunov and Bertram (2014),* is a 3D linear Rankine singularity method (RSM), solving the linear seakeeping problem in the frequency domain. The method was developed with focus on high-speed ships with large bow flare and stern overhang, for which the steady waterline strongly depends on the ship wave and the dynamic trim and sinkage. This fast code incorporates the effect of shallow water on the roll motion.

Analyses and software tools employed are in line with Second-Generation Stability Criteria Level 3 (Direct Stability Assessment), *IMO (2020)*. High accuracy of the simulations has been proven by intense validation against model tests and in-service measurements. The analyses assess synchronous and parametric rolling, but also direct excitation or loss of stability on wave crests, Fig.4. Ship motions are simulated in irregular, short-crested waves. For the case that a vessel has DNV class, the results for roll angle (and implicitly roll acceleration) are coupled with DNV rules for lashing load limits to assess criticality of situations. Otherwise, the maximum expected roll angle is compared against a maximum allowable roll angle.

The preprocessing analysis covers a densely populated simulation matrix including the complete range of loading conditions with respect to draft, trim and metacentric height, the full range of speeds, relevant water depths and the complete matrix of sea states according to IACS recommendation 34. Due to the large number of simulations, the analysis typically takes more than three weeks on a computer cluster. We deem this effort is required to maximize accuracy and subsequently minimize the number of false alarms. Results are stored in a database, for subsequent rapid interpolation in the apps.

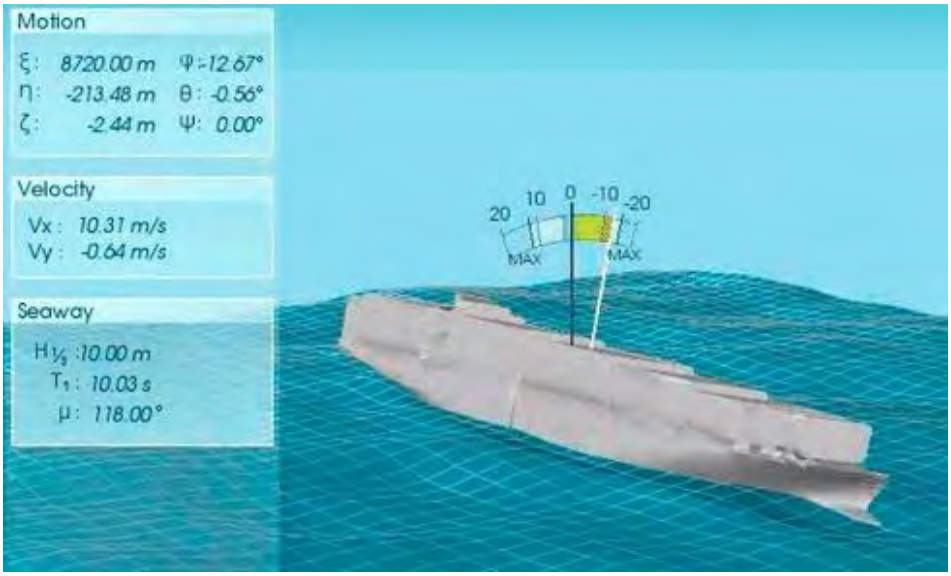


Fig.4: Nonlinear roll simulation

2.2. Anti-Roll Assist

For decision support from the Anti-Roll Assist app, the crew can enter the observed sea state to the application to evaluate the current condition and find immediate mitigation options, or they may enter predicted sea states for route planning. The required manual input is clustered by parameter groups, Fig.5:

- Operating condition: mean draft, trim, metacentric height, heading, speed, water depth
- Sea state: significant wave height, peak/mean wave period, mean wave direction for wind waves. Swell is optional and can be defined with values for the same quantities.
- Lashing: RSCS (Route specific container stowage) load reduction factor (if applicable) as prescribed by *DNV-RU-SHIP Pt.6 Ch.4 Sec. 10*

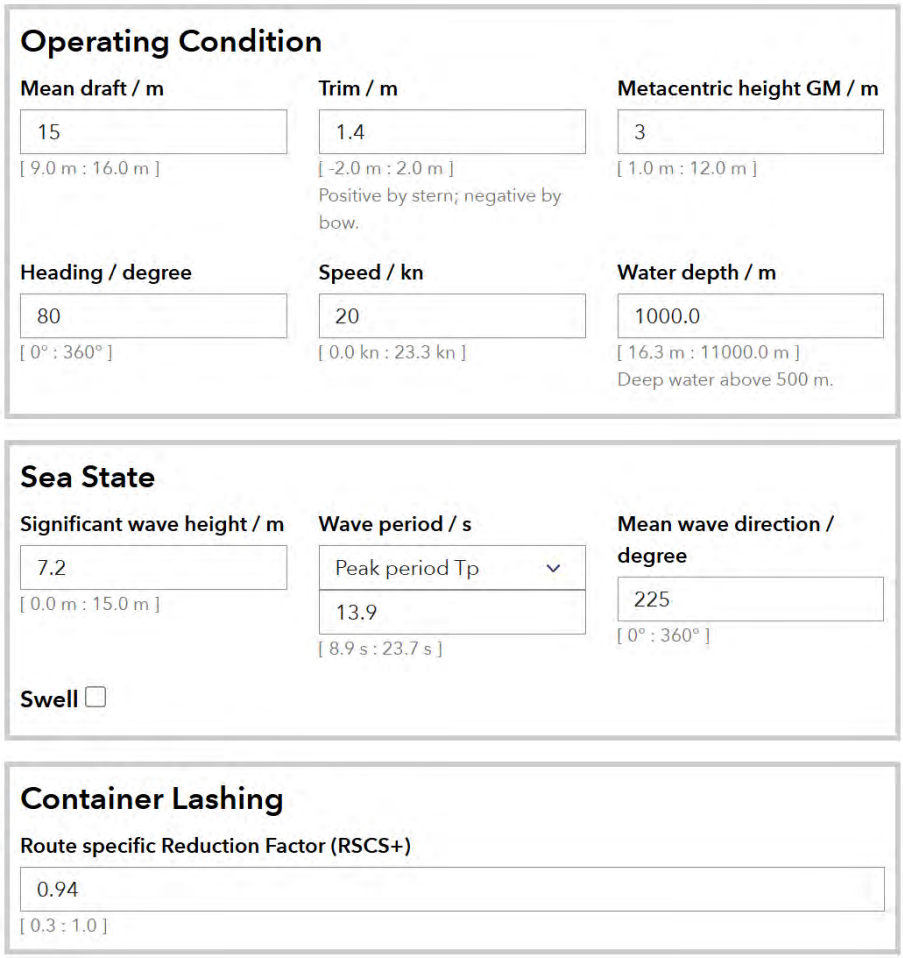


Fig.5: Anti-Roll Assist input

A risk scenario might be evaluated days ahead (e.g., for route planning), but weather may change quickly and timeframes from 3 to 6 h are recommended for higher accuracy in the weather forecast. If the risk picture (polar plot) shows a high risk of excessive rolling and reaching the design limits of the lashing system, the crew should consider the following options to avoid or reduce the risk:

- Change of heading
- Change of speed
- Change of GM with ballast water (considering global ship strength and stability)
- Change of course or speed early to avoid the area with unfavourable sea conditions

2.3. Case Study: Anti-Roll Assist

As an illustrative example, we can use the recorded track of a 14,000 TEU container ship sailing from Tokyo to Vancouver, Fig.6.

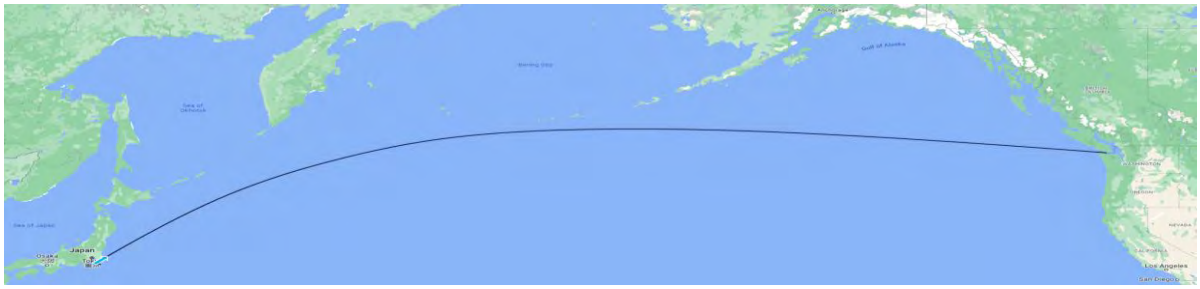


Fig.6: Planned route Tokyo-Vancouver of case study container ship

During the voyage, a low-pressure cell is coming down, Fig.7. The colours indicate the significant wave height. The weather forecast is made 24 h ahead. The ship will then be right in the storm area, Fig.8. The Anti-Roll Assist app will tell the crew whether the situation will be critical for the ship (and lashing systems) or not.

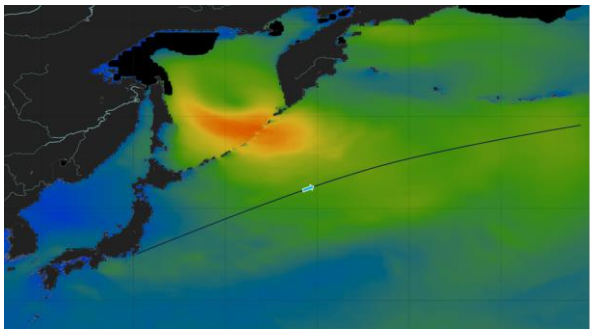


Fig.7: Large low-pressure zone approaching as ship is heading east

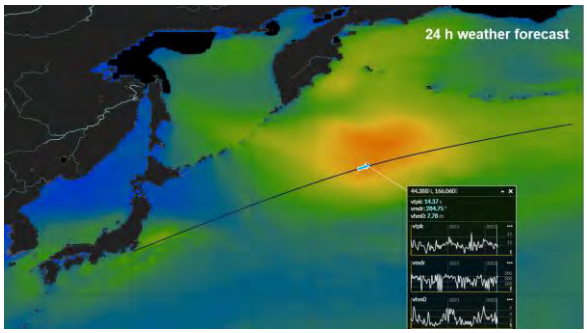


Fig.8: 24 h forecast for ship, predicting it will be in the storm

If the situation is critical, depends on the load condition for this ship. Fig.9 (left) shows the situation for GM = 5.0 m. There is no danger. Fig.9 (right) shows the same situation but for GM = 1.3 m. There is now a significant risk for parametric rolling in following waves. The warning from Anti-Roll Assist indicates, that the ship in the storm area can avoid the critical situation only by drastic changes of the course. However, the best approach would be to evaluate and react on the situation early, either by waiting 5 h for the storm to pass, or alternatively by changing course and sailing at the southern edge of the storm area where the waves and hence the roll motions will be substantially smaller.

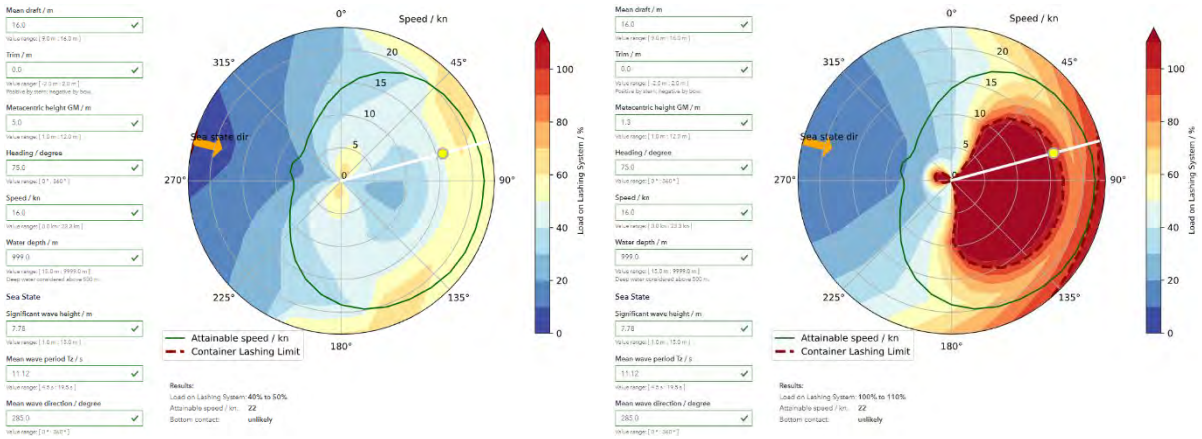


Fig.9: Anti-Roll Assist for GM = 5 m (left) and GM = 1.3 m (right)

Note that the numbers in this case study reflect industry reality and are representative for container ships of this size. For example, the “Maersk Essen” left Tokio with a GM of only 0.97 m, before it had a parametric rolling incident and lost 500 containers near Hawaii. Such cases must be identified and avoided beforehand.

2.4. Anti-Roll Assist API

Anti-Roll Assist requires considerable manual input. A more automated process would improve user-friendliness and allow easier interfacing with third-party applications. Therefore, we developed an API (Application Programming Interface), facilitating integration of Anti-Roll Assist into routing or navigation software solutions. Depending on the hosting system, there can be automated consideration of position, speed, heading, or weather forecast; and alerts can automatically be triggered.

Fig.10 shows a screenshot of StormGeo’s s-Suite. This routing system has navigational data and weather data available. Therefore, it can automatically check if there are situations along the planned route, where excessive loads on the lashing system become a problem. Fig.11 shows a screenshot of Navis’ Lashing Monitor software which integrates Anti-Roll with the lashing computation. More recently, Miros Group has combined Anti-Roll with their wave radar, displaying load on lashing system based on real time sea state measurements.

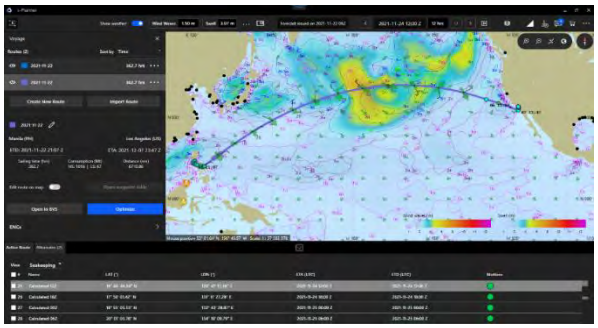


Fig.10: StormGeo’s routing software s-Suite

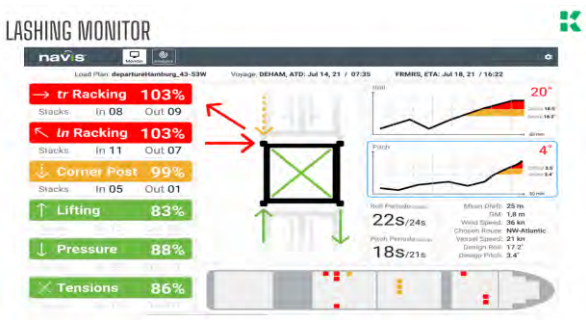


Fig.11: Lashing monitor software by Navis

2.5. Anti-Roll Alert

We believe there is business potential in a higher degree of automation and noticed that many customers do not have systems with all the required data onboard. Therefore, we decided to create a DNV solution, named Anti-Roll Alert.

Anti-Roll Alert requires the crew to only specify the loading condition and planned route at the beginning of a voyage. The system then works completely automated based on AIS data and weather forecasts and sends out alerts when a situation with high lashing loads is predicted. This reduces the user interaction significantly. If needed, the predicted voyage can be investigated in the user interface which can display the same polar plots as for Anti-Roll Assist for each point along the route, giving decision support for suitable course and speed changes.

Fig.12 shows an example of how Anti-Roll Alert supports crews. A 24,000 TEU container ship is near Cape Town, heading east. The service is tracking the vessel and retrieves weather information for all upcoming way points. The top right of the display shows what lashing loads are expected during the next four days. The two lines are the 100% limit and a user-specified alert limit. If a limit is reached, as in Fig.13, an automatic email alert is sent to the customer.

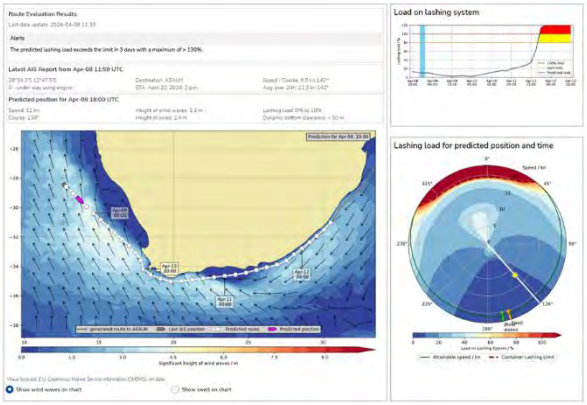


Fig.12: Ship in uncritical area

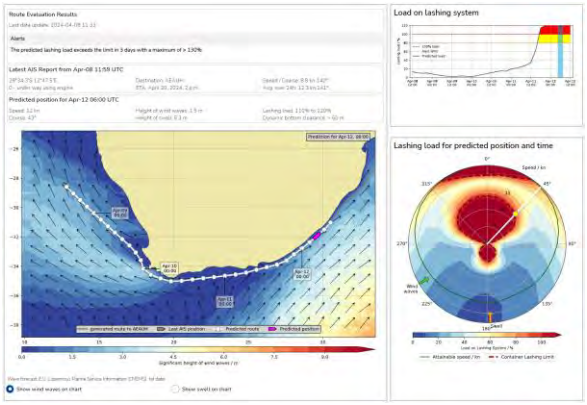


Fig.13: Ship in critical area → alert email is sent

3. Conclusions

Synchronous or parametric rolling occurs suddenly and may lead to expensive container loss. DNV's suite of decision support apps allow crews to recognize and avoid the risk before it becomes a threat. The apps are easy to use, are based on extensive seakeeping simulations of validated high accuracy and provide instant operational guidance.

Acknowledgements

We gratefully acknowledge the assistance of assorted colleagues in the development of the apps and the writing of the paper, namely Holger Jefferies and Arne Schulz-Heimbeck of DNV's CEC (Containership Excellence Center), Vladimir Shigunov of DNV's Advisory Ship Performance Center, and Volker Bertram of DNV's Maritime Competence Learning and Academy.

References

BERTRAM, V. (2012), *Practical Ship Hydrodynamics*, Butterworth & Heinemann

BIRAN, A.; LOPEZ-PULIDO, R. (2024), *Ship Hydrostatics and Stability*, Butterworth-Heinemann

IMO (2020), *Interim Guidelines on the Second Generation Intact Stability Criteria*, MSC.1/ Circ.1627, Int. Mar. Org., London

IMO (2023), *Explanatory Notes to the Interim Guidelines on the Second Generation Intact Stability Criteria*, MSC.1/Circ.1652, Int. Mar. Org., London

SHIGUNOV, V. (2009), *Operational Guidance for Prevention of Container Loss*, 10th Int. Conf. on Stability of Ships (STAB), St. Petersburg

SHIGUNOV, V.; BERTRAM, V. (2014), *Prediction of Added Power in Seaway by Numerical Simulation*, 9th HIPER Symp., Athens, http://data.hiper-conf.info/Hiper2014_Athens.pdf

SHIGUNOV, V.; EL MOCTAR, O.; RATHJE, H. (2009), *Conditions of Parametric Rolling*, 10th Int. Conf. on Stability of Ships (STAB), St. Petersburg

SHIGUNOV, V.; EL MOCTAR, O.; RATHJE, H. (2010), *Operational Guidance for Prevention of Cargo Loss and Damage on Container Ships*, Ship Technology Research 57/1, pp.8-25

SÖDING, H.; SHIGUNOV, V.; ZORN, T.; SOUKUP, P. (2013), *Method rolls for Simulating Roll Motions of Ships*, Ship Technology Research 60/2, pp.70-83,

SÖDING, H.; VON GRAEFE, A.; EL MOCTAR, O.; SHIGUNOV, V. (2012), *Rankine Source Method for Seakeeping Predictions*, 13th Int. Conf. on Ocean, Offshore and Arctic Eng. (OMAE), Rio de Janeiro

Appendix: Synchronous and parametric roll

The two resonant roll motion phenomena, which imply the potential for container loss, are "synchronous rolling" and "parametric rolling".

Synchronous rolling, Fig.A.1, *Bertram (2012), Biran and Lopez-Pulido (2024)*, happens when

- the wave direction is approximately perpendicular to the direction of travel (beam sea),
- the ship's natural roll frequency is close to the wave encounter frequency, and
- the metacentric height GM, which indicates the ship's initial static stability, is relatively high.

The effect is aggravated in shallow water, where the waves are shorter and steeper. This increases the risk of synchronous rolling disproportionally.

Fig.A.1: Synchronous rolling, *IMO (2023)*

Parametric rolling, *Biran and Lopez-Pulido (2024), Shigunov et al. (2009),* happens when:

- the wave direction is approximately aligned to the direction of travel ((oblique) following sea or head sea),
- wave lengths projected on ship centre plane is close to the ship length,
- GM is relatively low,
- the ship is slender with significant flare at the ship ends (waterplane and righting lever curves change periodically between ship-on-crest and ship-on-trough conditions), Fig.A.2,
- the encounter frequency is nearly twice the ship's natural roll frequency.

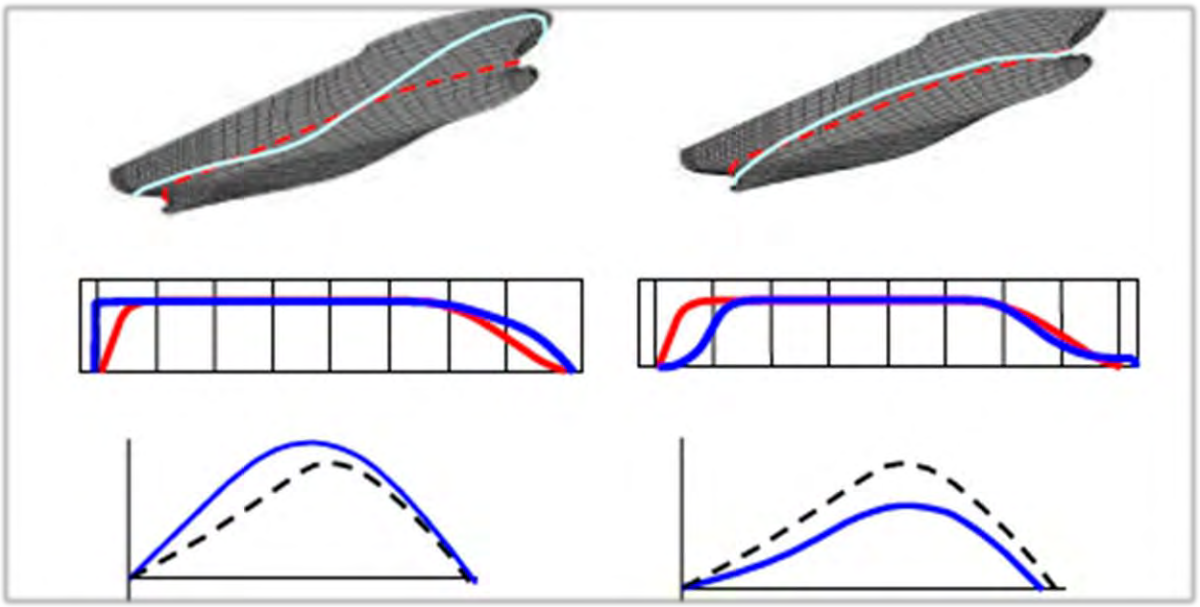


Fig.A.2: Waterplane area and stability curves in ship on wave trough (left) and on wave crest (right), *IMO (2023)*

The ship gets a push in the wave trough, does not feel much stability on the wave crest, and gets another push from the other side in the next wave trough, Fig.A.3. Then the roll angle can build up very quickly, within minutes.

Fig.A.4 illustrates the two types of rolling superimposed on photos taken on real ships.

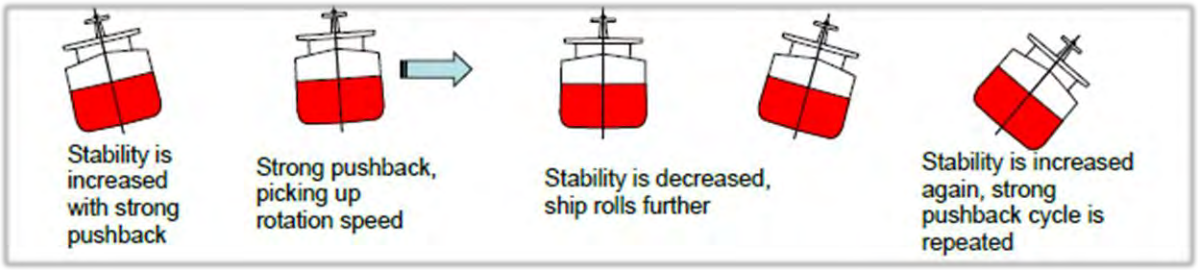


Fig.A.3: Roll angle builds up in parametric rolling, *IMO (2023)*

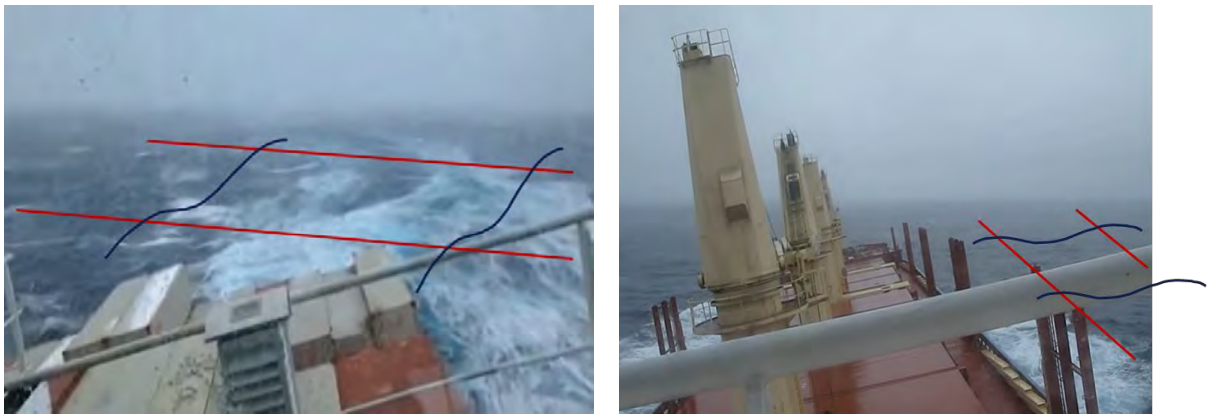


Fig.A.4: Parametric roll (left) and synchronous roll (right)

Two-Stage Conversion of GA Drawings into 3D Model Using Deep Learning

Jisang Ha, NTNU, Ålesund/Norway, jisang.ha@ntnu.no
Henrique Gaspar, NTNU, Ålesund/Norway, henrique.gaspar@ntnu.no

Abstract

This study explores the conversion of 2D General Arrangement (GA) drawings of ships into 3D arrangement models using deep learning. We start investigating the current status of key AI/deep learning tools available, such as ChatGPT, YOLO, DETR, and SAM, for our case. Our main study is performed using SAM, by a two-stage process: firstly, performing segmentation and extracting data from GA images. Secondly, combining data into a 3D model of a ship. Compared to the other tools or methods, which failed to function properly and had limitations, our approach successfully operated in recognizing GAs and extracting data. The approach was validated using GAs of commercial vessels, especially tankers. These results are expected to improve the utilization of 2D drawings currently used in shipyards and increase the connectivity and integration between early-stage ship design and later stages of work.

1. 3D from 2D GAs and the role of reverse engineering a hull

Reverse engineering is understood in the broader engineering and product development domains as the process of deconstructing an artifact - whether a machine, structure, or document - to extract design information, *Hess (2022)*. For the sake of our paper, the scope is narrowed as the methodology of extracting design knowledge from existing documentation when the original models are not available, *Legaz and Gaspar (2024)*. In t the context of this paper, reverse engineering does not necessarily rely on physical measurements or 3D scanning, but rather on analyzing technical drawings, such as General Arrangement (GA).

This approach is particularly relevant for academic research, where GAs published in sources like RINA's Significant Ships series, *RINA (2019)*, or preserved in archives as scanned drawings, often represent the only accessible design information. In the case of ships' GAs, the process begins with extracting geometric and arrangement information directly from PDF files or scanned images. The goal is to reconstruct essential parameters, such as principal dimensions, compartmentation, or equipment layout, that enable further analysis.

Unlike 3D scanning, which captures the geometry of physical objects, GA reverse engineering relies on careful interpretation of scaled drawings, annotations, and available metadata. Through digital processing - vectorization, scaling, or CAD re-drawing - these documents can be transformed into usable models. Such reconstructed models support academic studies, benchmarking, and comparative analyses, particularly when investigating historical vessels or ships where design data is otherwise inaccessible.

The aim of our incipient research is to develop a AI-supported re-engineering process able to convert hull from 2D GA to a 3D model. This is expected to significantly assist designers by integrating the fragmented 2D drawings and 3D models that exist at each stage of ship design. Specifically, it expects to extract the ship's hull as a 3D model from a 2D low resolution GA drawings.

We tested some of the the current AI tools before committing to SAM (Segment Anything Model). The first approach was an LLM, represented by ChatGPT. Drawing recognition via LLM worked successfully only for the text when a separate text layer existed in the drawing, and it still had difficulties understanding context when images and text coexisted. The second approach was a method of using image recognition models, commonly employed for drawing recognition. This approach required a very large amount of drawing data as a training set to address the domain shift. However, ship drawings have

relatively scarce training data; therefore, they showed low performance. After these attempts, we concluded that an approach based on segmentation models achieved the highest performance in ship GA. We performed segmentation and extracted data from GA images using SAM. Then, we involved combining the extracted data into a 3D model of a ship as the next stage. Through this approach, we were successfully able to perform the conversion from 2D drawings to 3D models. This is presented as follows.

2. Current Limitations on the Recognition of GA Using LLM and Image Recognition

2.1. Using LLM for Recognition of GA

For the conversion of 2D General Arrangement (GA), we can firstly consider utilizing Large Language Model (LLM) to recognize GA. Fig.1 shows the result of requesting a mock-up version of the GA of NTNU's research vessel "Gunnerus" to be input and reproduced as a 2D graph. (ChatGPT 4o) "Analyze the differences between the two designs."

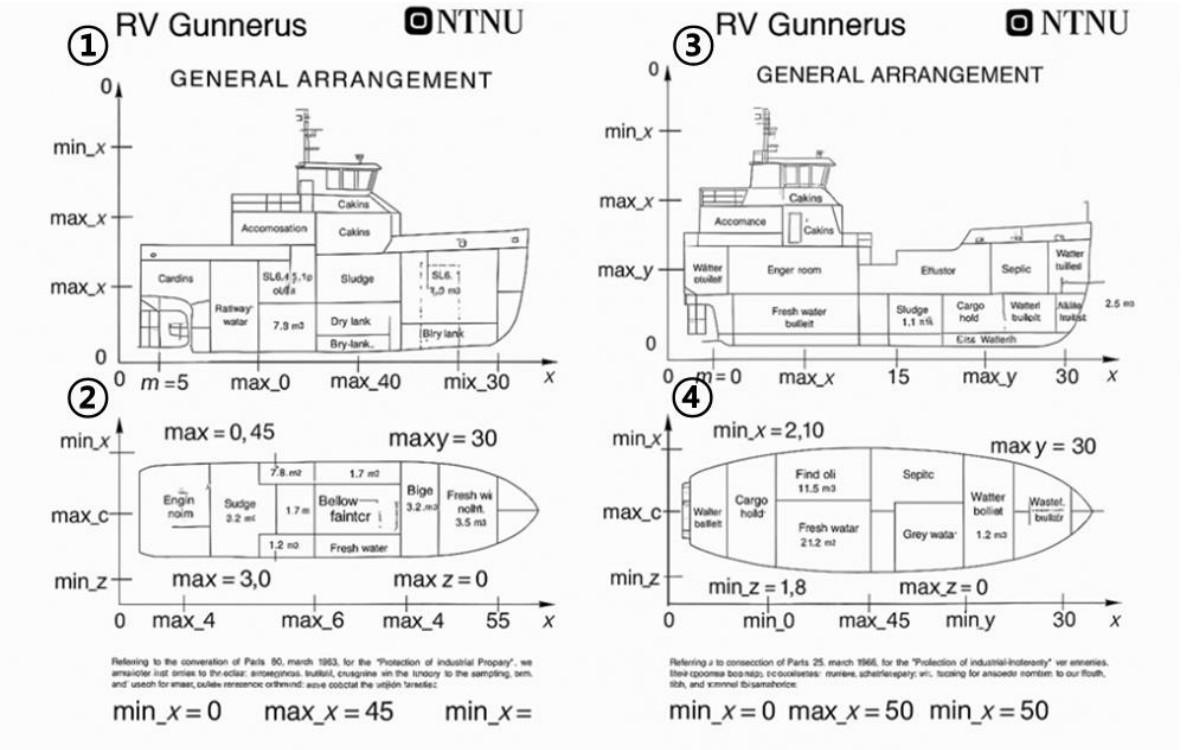


Fig.1: Results of GA recognition by LLM (ChatGPT 4o)

In Fig.1, ChatGPT failed to recognize several information. In Fig.1(1), max_40 appears before min_30, and other figures also show similar issues. Furthermore, it often confuses certain information with the names of arrangements. Other LLMs, such as llama 3.2, also show similar results. However, results obtained using only LLMs often fail to consider the text written on the image (drawing) or mistakenly identify the text as part of the image. Fig.2 is an example of extracting only the text layer from the same drawing and re-extracting only the ship's arrangement and hull information.

If the drawing is in a file format such as PDF or DWG that has a separate text layer, the required data could be successfully extracted as shown in Fig.2. However, the problem is that even if designers use the appropriate format, they often fail to separate all text layers. Furthermore, layers frequently become fragmented when handled by multiple designers. Therefore, recent attempts have been made to improve recognition rates when images and text coexist by utilizing LLM and text recognition algorithms (such as OCR technologies) together. *Chen (n.d.)* attempted to improve results by utilizing Tesseract OCR (Optical Character Recognition) on some images, such as receipts, rather than using LLM alone.

Design Comparison Table: RV Gunnerus (Before vs After Rebuild)

Space	Old Design Location/Size (1463114)	New Design Location/Size (165821B6)	Change Summary
Engine Room	Aft lower section (below 1st deck), approx. *9.60×4.20* m (≈31.3 m²)	Similar position, slightly extended aft, almost same size	Minimal change in size. Slightly shifted aft due to hull extension.
Sludge Tank	Near engine room, 1.7 m³	Same location, 1.7 m³	No change in volume.
Bilge Tank	Bottom of engine room, 3.5 m³	Same location, 3.5 m³	No change in volume.
Fresh Water Tank	Midship (below), 11.2 m³	Midship (expanded section), 17.0 m³	Volume increased by approx. +5.8 m³. New tank or expansion added in lengthened area.
Black/Grey Water Tanks	Black: 3.2 m³, Grey: 3.2 m³	Same: Black 3.2 m³, Grey 3.2 m³	No change in volume or location.
Oil Waste Tank	Near engine room, 1.8 m³	Same: 1.8 m³	No change in volume.
Cargo Hold	Mid-upper section, 21.2 m³	Mid-upper section, 21.2 m³	Same volume. Slight aft shift due to hull lengthening.
Ballast / Roll Damping Tanks	Ballast only: 47.8 m³ total (No.12: 5.0 + No.13sb: 5.0 + No.13p: 37.8)	Same ballast capacity: 47.8 m³ + **New Roll Damping Tank: 21.2 m³**	**Roll damping tank added.** Ballast tank sizes unchanged, but geometry adjusted.
Bridge / Control Room	Superstructure center top (approx. 15 m²)	Same area, shifted slightly aft	Same function and size. Repositioned aft with new superstructure layout.
Accommodation / Labs / Conf. Room	A-deck: Conf. Room 20 m², others not precisely listed	A-deck: Conf. Room 20 m² + **New Lab (Dry Lab 8.5 m²)**, tech rooms	Existing conference room kept. **New technical/lab areas added** such as Dry Lab (8.5 m²).

Fig.2: Extracted data from the text layer of the drawing

In this study, considering the characteristics of ship's GA, which consists of a mixture of various images, texts, and information, we aim to recognize the components of GA and extract information by attempting various types of image recognition deep learning technologies.

2.2. Using Image Recognition Models for Recognition of GA

There are several representative models commonly used for image recognition, e.g., YOLO and DETR. YOLO (You Only Look Once), *Redmon et al. (2016),* is a model for image recognition that was first proposed in 2016 and has evolved into more than 12 different versions, including YOLOv12. DETR (Detection Transformers), *Carion et al. (2020)*, is an image recognition model proposed by Facebook AI Research that utilizes a transformer-based architecture to achieve good results in image recognition. The following figure shows the results of attempting ship hull recognition or arrangement recognition using image recognition with different models.

Image recognition results using YOLO and DETR for GA drawing did not perform consistently well, Fig.3. It failed to recognize intended objects or the ship hull, and even after setting new classes and training the model, we observed that the original classes were recognized incorrectly more frequently. This is the domain shift problem that occurs when applying such AI models to a different target domain with which they were trained, and it arises when tuning or retraining with a small number of images. For image recognition approaches, accumulating a larger dataset of over 10,000 images is expected to be necessary to extract meaningful data from GA for constructing 3D models. While studies have utilized various models beyond the one we present, *Stensrud and Klausen (2022)*, many are attempting to recognize drawings using new approaches. Segmentation models are one of the novel approaches

that performed better than image recognition approaches when targeting ship drawings. The results of tuning and training the best-performing model, SAM 2.1, *Ravi et al. (2024)*, are introduced in the next chapter.

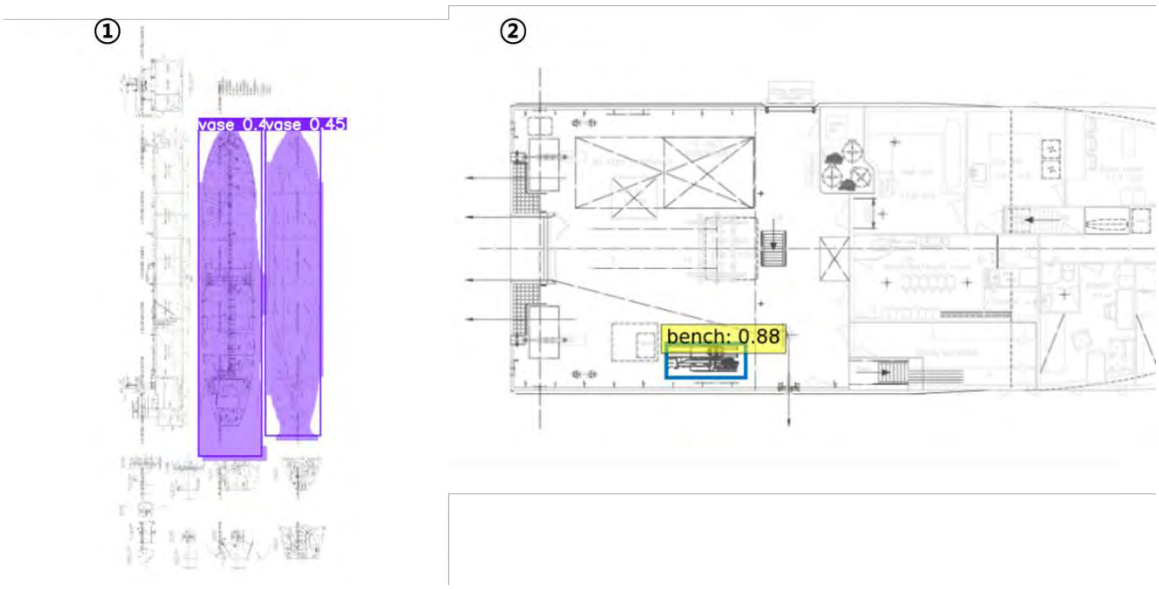


Fig.3: Image recognition results by YOLOv12x (1) and DETR (2)

3. Two-Stage Conversion of GA Drawings into 3D Model Using Deep Learning

3.1. The Process of Two-Stage Conversion of GA Drawings into 3D Model

In this section, we introduce our process of two-stage conversion of GA drawings into 3D model using deep learning techniques. Fig.4 illustrates the entire process.

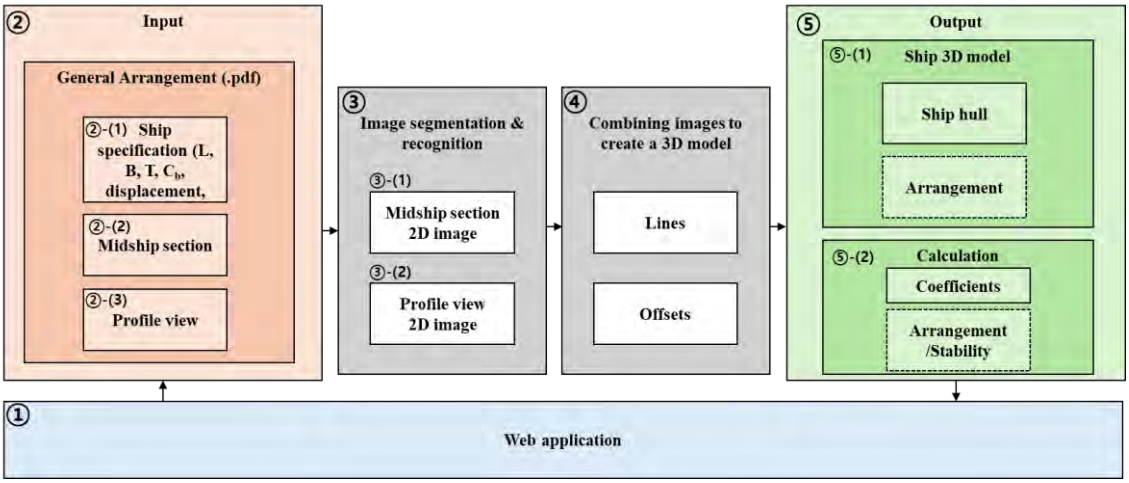


Fig.4: The process of two-stage conversion of GA drawings into 3D model

The procedure is represented in Fig.4, in which each number represents the sequential order in which each process is executed, and the dotted lines indicate functions that are planned but not yet implemented. Throughout this entire application, we connected and displayed all input and output results through the web application, Fig.4(1). As input, we used publicly available GAs from digital versions of *RINA (2013-2018)*. We used the information on the ship specifications (task 2-(1)) within GA and the figures for the midship section (task 2-(2)) and profile view (task 2-(3)) as input for our model.

Fig.4(3) illustrates the first stage of the conversion. The first stage recognizes the hull from GA drawings using image segmentation. To achieve this, the study separated and processed the midship section drawing and the ship longitudinal profile drawing for recognition and segmentation. For image recognition and segmentation, various published models such as YOLO, DETR, and SAM were tested. Among these, SAM 2.1, which showed the best results, was selected and trained using GA. We separated the midship section image (task 3-(1)) and profile view image (task 3-(2)) into distinct datasets for training to achieve better results, extracting information from each drawing.

The second stage, Fig.4(4) involves creating a 3D model based on the results from the first stage. By expanding the ship's shape identified from the midship section image along the lines obtained from the profile drawing, a ship 3D model was generated with lines and offsets.

As output, we created a 3D model of the ship hull accessible via a web application and calculated the vessel's basic volume and coefficients for user verification.

3.2. Recognition of Midship Section

We start with task 3-1 using the SAM2.1 model, which proved most effective at segmentation of hulls in GA drawings. The results of the SAM2.1_large model's recognition of the midship section, performed without any training or tuning, are shown in Fig.5.

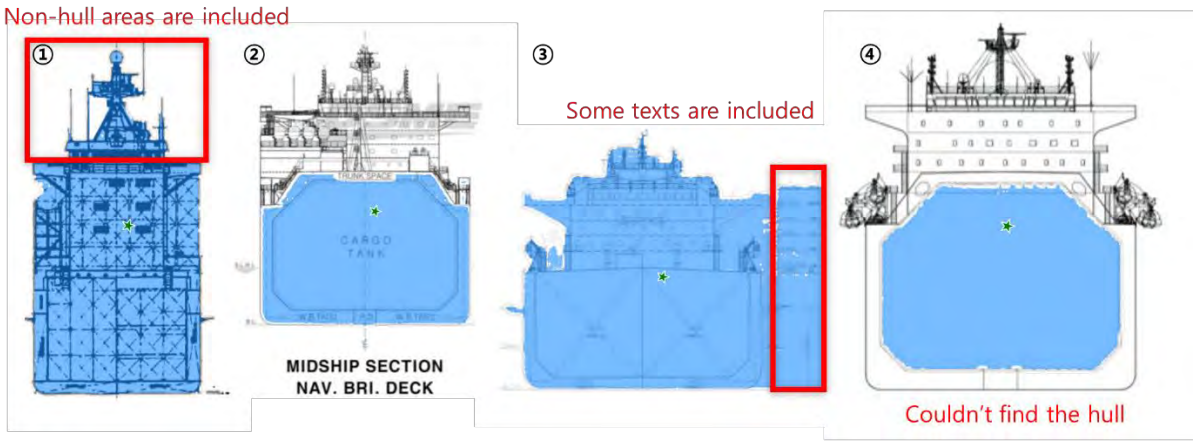


Fig.5: Results of segmentation and recognition of ship hulls from midship section drawings (Before training)

In Fig.5, we input the drawing image into the model as input, and the area rendered in blue represents the region the model predicted to be the ship hull. We highlighted certain characteristic errors in red. Fig.5(1) successfully performed segmentation, but since it was performed without superstructure knowledge and learning about ship hulls and other parts, it recognized all figure areas as a single object. Fig.5(2) performed relatively successfully but still failed to recognize some areas properly. Fig.5(3) shows the same problems as those found in LLM and other networks. When text explaining the figure is present around it, the text is often recognized as part of the figure or as a single object. In the case of Fig.5(4), the model confused tanks or arrangements inside the ship hull with the ship hull. The results of 34 GAs are summarized in the Table I.

Table I: Summary of the results of midship section (Before training)

Max. Width error (px)	Max. Height error (px)	Average error of width	Average error of height
238	151	36.6%	37.9%

The following is an example of the segmentation results obtained by performing segmentation using a model trained with RINA's data.

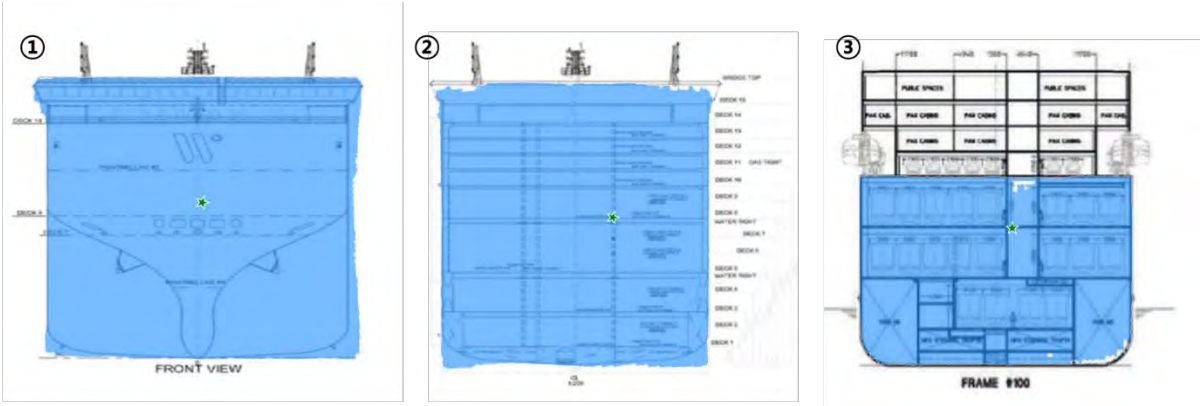


Fig.6: Results of segmentation and recognition of ship hulls from midship section drawings (After training)

In Fig.6, all the issues encountered in Fig.5 have been resolved. Segmentation was performed only on the ship hull, excluding superstructures. Fig.6(2) showed good results despite containing both figures and various texts. Fig.5(3) also successfully excluded the superstructure. The summarized results after the training are in Table III.

Table II: Summary of the results of midship section (After training)

Max. Width error (px)	Max. Height error (px)	Average error of width	Average error of height
14	27	1.70%	2.61%

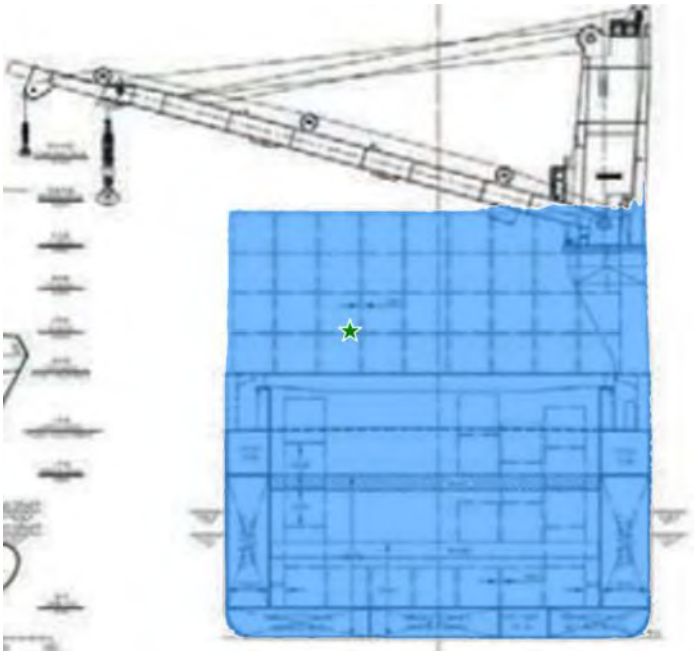


Fig.7: An example of recognition of midship section

Problems still occur in scenarios like Fig.7. When a grid for the crane or cargo is drawn in the midsection, the model sometimes misinterprets it. The errors in width and height are the result of this, and excluding these examples has a similar error rate with width, approximately 1.71%.

3.3. Recognition of Profile View

For task 3-2, the SAM2.1_large model, as described in Section 3.1 and Fig.4, was applied to the profile view of the ship GA. The number of ship GAs used for training is the same as task 3-1 (in Section 3.2).

An example of the results is shown in Fig.8.

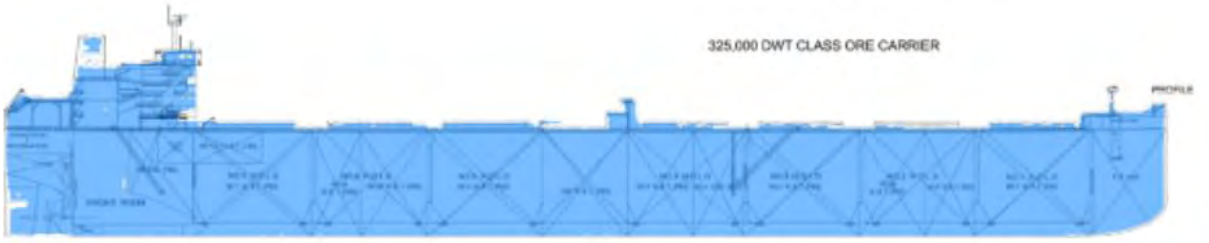


Fig.8: An example of recognition of profile view

Ship hulls are generally well recognized and can be segmented by GA, but accuracy tends to decrease when multiple superstructures are found. The summarized results after the training are in Table III.

Table III: Summary of the results of profile view (After training)

Max. Width error (px)	Max. Height error (px)	Average error of width	Average error of height
130	1308	34.7%	40.6%

Despite showing good results in Fig.8, the poor performance in Table III is due to our current model being trained with a bias toward specific ship types. While it achieves very high accuracy for ship hulls like tankers, it performs poorly with high error rates for passenger ships or other hull types.

4. Application Using the Proposed Method

We created a 3D model as task 4, using the method proposed in Section 3, with the midship section view and profile view of the ship GA as input. The target vessel is an ore carrier from *RINA (2018)*. Figs.9 and 10 show the results of recognizing the vessel's GA and representing it with points and lines.

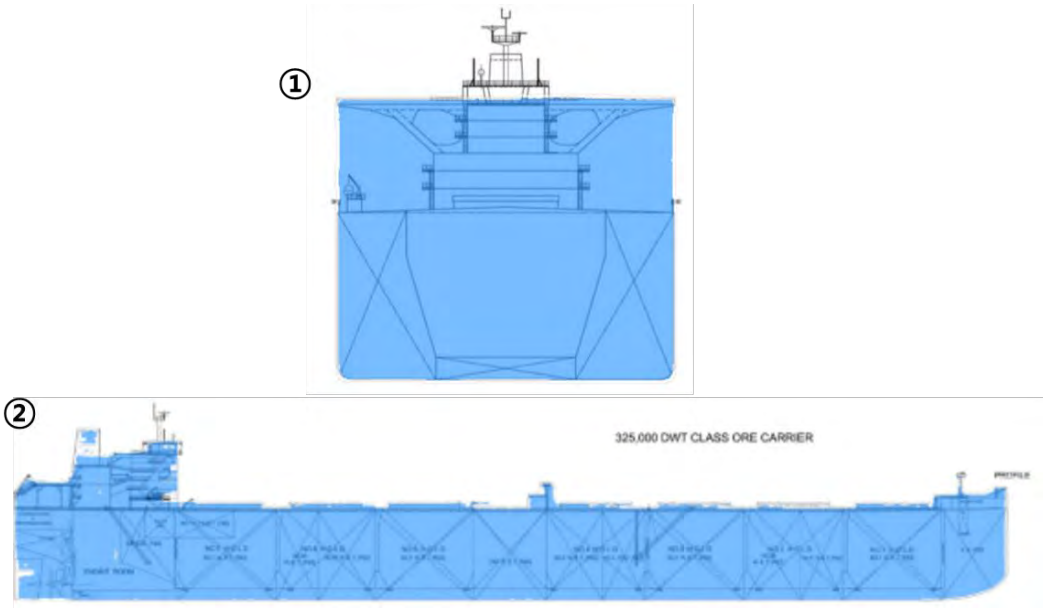


Fig.9: An example of recognition of GA

The midship section, Fig.9(1), and profile view, Fig.9(2), images were input, with the blue area representing the point predicted as the ship hull by task 3. Representing this again with more refined points and lines are in Fig.10.

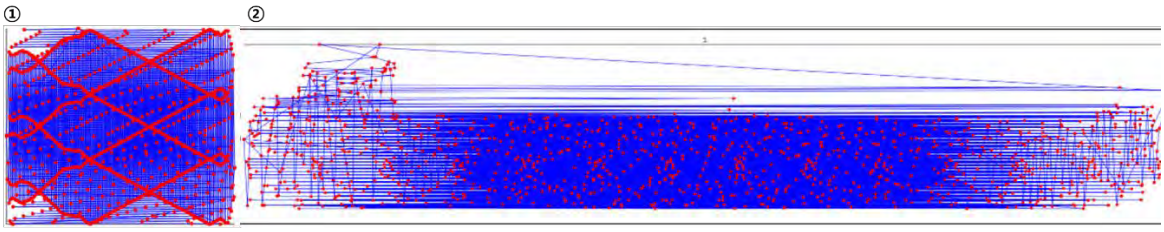


Fig.10: An example of recognition results expressed as points and lines

For Task 4, we extracted only the ship hull portion below the draft from the results of Task 3 (Figs.9 and 10), expanded it into 3D, and connected the pieces. The result is shown in Fig.11.

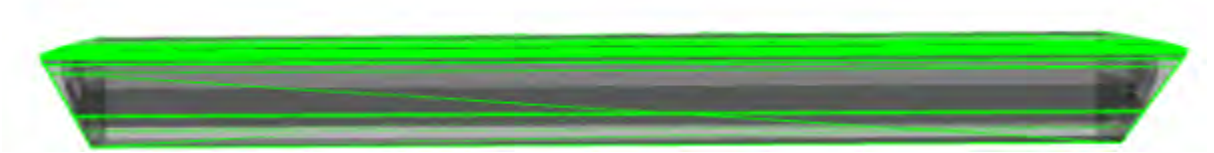


Fig.11: An example of hull 3D model from GA

Based on the model obtained, displacement and the block coefficient (C_b) are estimated. The comparison results are shown in Table IV. The ship's C_b could be estimated with an error of approximately 3.9%. However, this estimation is performed on a ship with a large C_b and a simple hull form. For ships of various hull types, the accuracy can be potentially lower.

Table IV: Summary of the results of a 3D model

	Displacement (ton)	C_b
Actual ship	453,463.2	0.8323 @23.0
Generated model	435,865	0.80

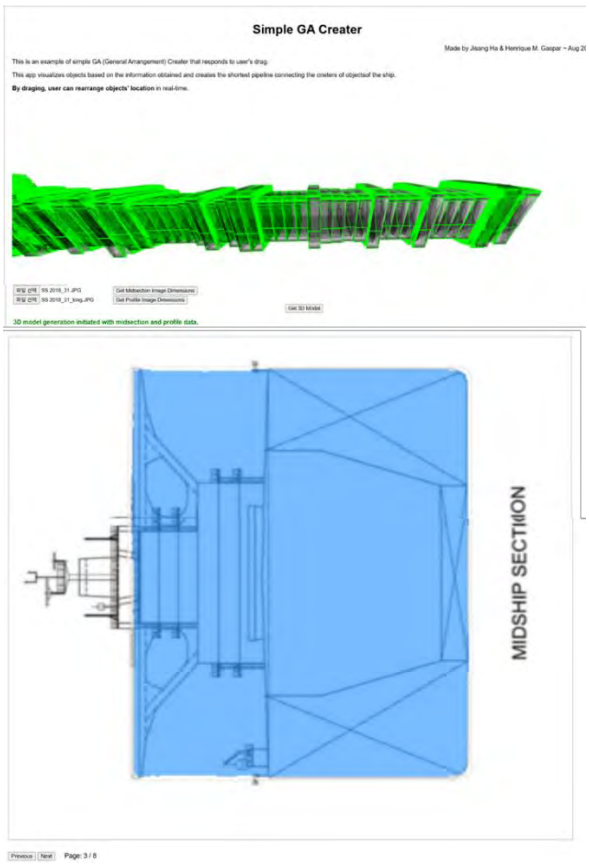


Fig.12: An example of the web application

Fig.12 shows an incipient example of the configured web application. (At the time of the submission of the article, the web application is being developed, therefore the irregular 3D version observed. We expect to have a smoother version running by the day of the conference.) When the GA drawing is entered as input, the 2D segmentation results for that figure are displayed at the bottom according to the page, and the 3D model is generated and displayed at the top.

5. Conclusions

Reverse engineering of GAs through computer vision provides a practical method to extract editable geometries when original design models are unavailable. This process can also be extended to onboard applications, where visual data capture may substitute for additional sensors.

Progress in this area will depend on collaborative efforts between academia, industry, and regulatory bodies to establish reliable methods, evaluate limitations, and define standards. Attention to cybersecurity will remain important, since the same technologies can be misused if not properly safeguarded. We suggest the same approach as *Legaz et al. (2025)*: training the model right while training the right model.

Training the Model Right: When applying computer vision and AI techniques to reverse engineer ship GA drawings, the quality of training data and the process of model development are decisive. Unlike physical 3D scans, GAs exist as technical documents with varied formats - ranging from vector-based PDFs to scanned raster images. Training the model right therefore requires datasets that capture this diversity and reflect the specific conventions of naval architecture drawings. Without such domain-specific data, generic computer vision models risk misinterpreting or omitting critical features. Open datasets of ship drawings, coupled with transparent annotation protocols, can help establish a foundation for reliable feature extraction. Expert validation is paramount, ensuring that the AI correctly identifies ship compartments, machinery spaces, and arrangement details in line with established design practices.

Training the Right Model: Beyond training quality, it is also necessary to ensure that the models themselves are suited to handle GAs as use case. Models developed for generic document analysis or architectural plans may not account for the conventions and standards of ship design. Training the right model involves tailoring architectures and workflows specifically for maritime drawings. This includes incorporating recognition modules for ship-specific symbols, multi-scale analysis to handle both global dimensions and fine details, and alignment with classification schemes used in ship registers and regulatory documents. Validation against reference vessels and cross-checks with hydrostatic or stability data can ensure that the extracted GA information is not only geometrically consistent but also meaningful for subsequent analyses.

In the case of GAs, progress will require both strategies: training the model right, with carefully curated and annotated datasets, and training the right model, with methods adapted to the characteristics of ship design drawings.

Moreover, there is still potential for error correction in the process of building the 3D model by synthesizing the information. Furthermore, while the current process requires manual separation and input of the GA, future research aims to enable the construction of a 3D model solely by inputting the GA through combination with LLM models.

The results are expected to improve the utilization of 2D drawings currently used in shipyards and increase the connectivity and integration between early-stage ship design and later stages of work with the automation of GA reverse engineering and support reliable reuse of legacy documentation in maritime research and education.

References

CARION, N.; MASSA, F.; SYNNAEVE, G.; USUNIER, N.; KIRILLOV, A.; ZAGORUYKO, S. (2020), *End-to-End Object Detection with Transformers*, https://github.com/facebookresearch/detr

CHEN, J. (n.d.), *Enhancing Image Text Extraction with LLM and OCR*, https://medium.com/@jameschen_78678/enhancing-image-text-extraction-with-llm-and-ocr-8221cb555cc5

HESS, B. (2022), *What is reverse engineering and how does it work?*, Technical report, Astromachine Works, https://astromachineworks.com/what-is-reverse-engineering/

LEGAZ, M.J.; GASPAR, H.M. (2024), *Computer vision for reverse engineering in the design, simulation and operation of maritime systems*, European Council for Modelling and Simulation, ECMS, 38(1), pp.234–241

LEGAZ, M.J.; GASPAR, H.M.; ICHINOSE, Y. (2025), *Open software for teaching and research maritime design and engineering*, ECMS, https://doi.org/10.7148/2025-0039

RAVI, N.; GABEUR, V.; HU, Y.-T.; HU, R.; RYALI, C.; MA, T.; KHEDR, H.; RÄDLE, R.; ROLLAND, C.; GUSTAFSON, L.; MINTUN, E.; PAN, J.; VASUDEV ALWALA, K.; CARION, N.; WU, C.-Y.; GIRSHICK, R.; DOLLÁR, P.; FEICHTENHOFER, C.; FAIR, M. (2024), *SAM 2: Segment Anything in Images and Videos*, 13th Int. Conf. Learning Representations (ICLR), pp.41175–41218, https://arxiv.org/pdf/2408.00714

REDMON, J.; DIVVALA, S.; GIRSHICK, R.; FARHADI, A. (2016), *You Only Look Once: Unified, Real-Time Object Detection*, pp.779–788, http://pjreddie.com/yolo/

RINA (2019), *Significant Ships of 2018*, The Royal Institution of Naval Architects, London

STENSRUD, E.; KLAUSEN, K. (2022), *Another Step towards Remote Inspections of Maritime Vessels using Tailored Inspection Drones Instrumented with Computer Vision*, https://utkilen.no/fleet/core-trade

Multi-Structure Product Data Management in Ship Design

Dawid Stade, CONTACT Software, Bremen/Germany, dawid.stade@contact-software.com
Maximilian Idjen, CONTACT Software, Bremen/Germany, maximilian.idjen@contact-software.com
Elisabeth Brandenburg, CONTACT Software, Bremen/Germany, elisabeth.brandenburg@contact-software.com

Abstract

In shipbuilding, documentation, plans, and drawings are typically the primary deliverables. Despite the absence of a unified "single ship model", digitalisation facilitates the connection of these documents to a digital product model. Traditional product data models are the outcome of the ability to organise components into assemblies, which is represented by a hierarchical structure. However, the varying demands throughout a ship's lifecycle and the complexity of the vessel necessitate extensions to this. Thus, an interconnected data model comprising spatial, system-oriented, and engineering-oriented structures is proposed to accommodate planning information and to offer different views on information objects for multiple disciplines. As it is validated on practical requirements, this enhances the shipbuilding process by providing a holistic representation of the ship.

1. Introduction

As in other industries, ship design is undergoing continuous digitalisation, dating back to the 1970s. Early developments, such as the adoption of computer-aided design, have culminated in the current era of smart digitalisation. Nowadays, naval architects and engineers are provided with tools and methods that enhance the efficiency of design and, moreover, elevate its quality over the entire life cycle. These encompass, for instance, the parametric generation and optimisation of hull forms, as well as simulation-driven ship design, *Papanikolaou et al. (2024)*.

The SEUS project, https://cordis.europa.eu/project/id/101096224, was initiated with the objective of promoting the digitalisation of shipbuilding. The overarching aim of the project is to establish a connection between the domains of design, simulation and optimisation applications for shipbuilding and the Product Lifecycle Management (PLM) realm. A European consortium of shipyards, CAD and PLM vendors, and research institutions is implementing a platform that integrates CAD and PLM processes, specifically designed for the shipbuilding industry.

Ship design is typically project-driven and tender-based. In comparison to other industries, batch sizes are commonly small. The sheer size of a ship poses a challenge, resulting in an enormous quantity of information that arises during design. For instance, for one icebreaker vessel, up to 60,000 documents and drawings, 100,000 issues and 150,000 components require storage, organisation and management. The biggest vessels, such as aircraft carriers, easily surpass 1,000,000 parts. These objects occur at different design phases and have their own life cycles. Ship complexity presents another key challenge, especially in European shipbuilding, which focuses on highly specialised vessels, *Kamola-Cieslik (2021)*. Ship design is confronted with diverse vessel types, numerous and varied requirements, a complex process of requirement elucidation, additional non-economic and non-operational demands, and ambiguous engineering responsibilities. Consequently, ships are classified as physically large and complex systems, with a design process being akin to civil engineering rather than to the development of smaller vehicles, such as cars, *Andrews (2013)*.

Van Den Hamer and Lepoeter (1996) delineate five dimensions of product data management: views and hierarchies, versions and statuses, as well as product variants. The former are also designated as a taxonomy, which comprises a hierarchical decomposition of the product data and corresponding perspectives. To address the aforementioned challenges in ship design, diverse perspectives are employed, contingent upon the specific design phase and domain. While, for example, functional views

are required in early ship design, production planning demands spatial views, dividing the vessel into zones in which the ship is to be manufactured, *Aragão Fonseca et al. (2023)*. Furthermore, multiple disciplines within the domains are dependent on updates and information from each other in real time. The corresponding hierarchical structures need to organise product data adequately.

A conventional taxonomy, such as that offered by the majority of Product Lifecycle Management (PLM) systems, hierarchically divides the product into assemblies and components. Its layers are reflected by relationships between the elements that can be designated as “is part of” or “consists of”. The standard taxonomy fails to satisfy shipbuilding demands. Therefore, this study aims to propose a data model capable of addressing the multifaceted requirements of the ship design process. Given the absence of standardised practices and the resultant heterogeneity in ship design implementations across European shipyards, the proposed model must also exhibit the flexibility to accommodate diverse procedural variations, *Bronson et al. (2024)*. The presented data model is implemented within and specifically for CONTACT Elements, a modular and flexible Product Lifecycle Management system.

2. Related Literature and Research

Ship design encompasses the entirety of a vessel's life cycle. The development process can be segmented into discrete phases, *Papanikolaou (2010)*. The process commences with the conceptual design phase, which *Andrews (2013)* describes as follows: Following a solicitation of bids by the prospective owner, the shipyard is required to respond within a constrained timeframe, occasionally as short as six weeks. The proposal must encompass a technically viable concept, detailed cost projections, and a development and construction schedule. The actual approach highly depends on the novelty of the ship being developed. However, in commercial shipping, shipyards' financial risk encourages evolutionary development. This involves exploration, concept studies, as well as concept design and selection. Initially, a preliminary and broadly distributed analysis is conducted to ascertain the feasibility of achieving the stipulated requirements. This usually results in several concepts in which questions of exploration are clarified in greater depth. In the subsequent concept development, decisions such as the selection of the ship style, equipment and performance characteristics are made iteratively. With the help of a synthesis model, the ship size and architecture are determined and evaluated. In this early phase, a functional view of the ship is often used, such as the Ship Work Breakdown Structure (SWBS) and the Skipsteknisk Forskningsinstitutt (SFI) Group system, *Bronson et al. (2024)*. They divide the ship into elements such as the hull structure, propulsion unit, power generation, auxiliary systems, equipment and furniture.

In the following contractual design phase, all necessary documents for the contract's conclusion must be prepared and agreed on. These include essential drawings, material and equipment lists, as well as detailed specifications. Additionally, a schematic representation of all systems is provided, including the main drive, pipeline systems, electrical systems, and cargo systems, *Misra (2015)*.

With the signing of the contract, the detailed design begins, in which, according to its name, a fine-grained definition and evaluation of the design take place. Furthermore, a detailed planning of the production of the ship takes place, for which, in addition to the aforementioned functional views, those that are suitable for the production of the ship come into focus, *Aragão Fonseca et al. (2023)*. For example, the ship is divided into grand blocks, decks, spatial objects and zones, depending on the discipline, each containing all manufactured and installed components within that area. An example of such a hierarchical view is the Product Work Breakdown Structure, *Pal (2015)*.

In the construction phase, individual steel plates are manufactured and assembled into blocks and grand blocks after preparatory measures. Before these are erected and welded for assembly to the entire ship, the most extensive assembly of the heaviest and largest components, such as the main engines, takes place, as this is comparatively easy in this state. After the composition of the grand blocks and blocks, the remaining equipment, including all modules and interior components, is installed.

During the operation, functional views are again in the foreground, offering a wealth of detail on systems, subsystems, components, modules and parts, *Aragão Fonseca et al. (2023)*. The precise identification of the equipment contained in the views is given, for example, by the aforementioned SFI Group System. With the reaching of the end of life of a ship, the decommissioning is in the foreground, which can consist of a refit, re-commissioning, sale or scrapping. In the case of scrapping, the product structure should list all those components that need to be decommissioned. In addition, information on toxic or hazardous materials must be included, *Andrade et al. (2015)*.

Among the approaches to cope with these requirements by a PLM system, the Fourth-Generation Design (4GD) should be emphasised. 4GD aims to combine an effective virtual design environment with comprehensive product data management. It consists of a flat product hierarchy consisting of design elements that organisational partitions can group. This approach allows different taxonomies to be mapped. However, as *Levišauskaitė et al. (2017)* report, this approach cannot achieve an effective improvement in the reuse of 3D models.

3. Core Elements of the proposed data model

Taking into account the size and complexity of a ship, different views on a ship's product data are required depending on the design domain and phase. These comprise functional and spatial views, each requiring a hierarchical data structure.

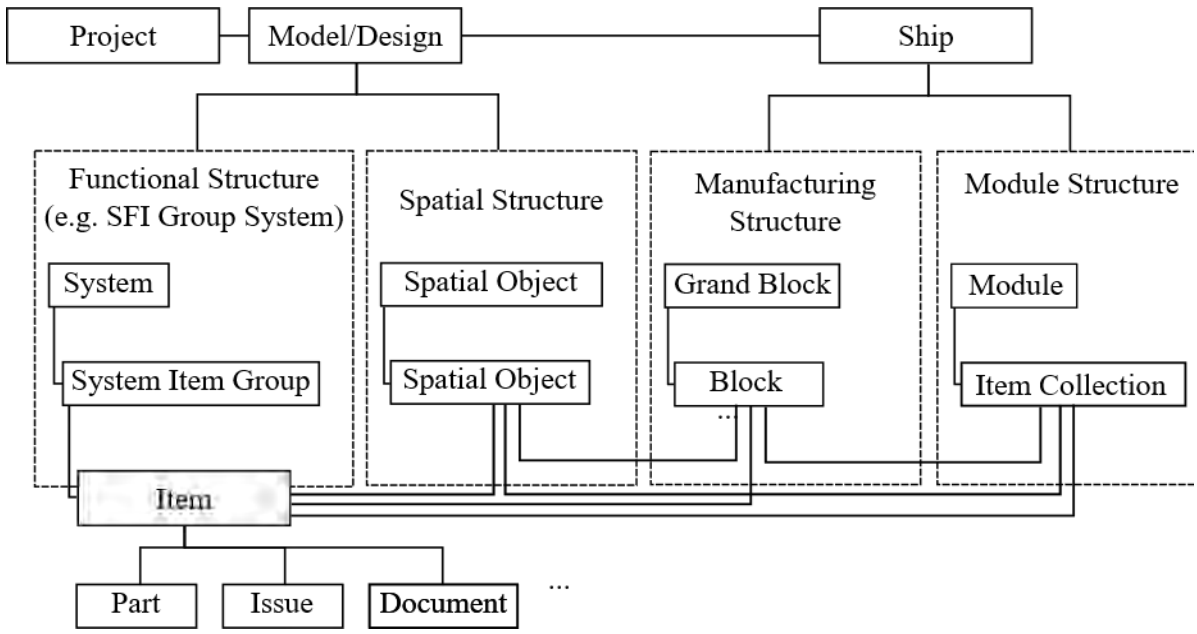


Fig.1: Core elements of the proposed shipbuilding PLM data model

Fig.1 shows an abstract representation of the structures, customisable for specific use cases. The functional structure depicted on the left can be implemented, for instance, as an instantiation of the SFI Group System. The spatial structure, along with the functional structure, can be subsumed under the overarching concept of model or design. Overall, the different uses of the structures highlight the overlap between shipbuilding and factory planning characteristics. The industry-neutral model emphasises the significance of functional and spatial structures in factory planning, providing the foundation for related perspectives. A design is employed in shipyards for the iterative development and construction of similar vessels. Consequently, a design functions as a template, encompassing critical spatial and functional architectures, incorporating varying levels of detailed information.

The right-hand side of the figure depicts structures exclusively employed during the instantiation of a model or design, specifically the development of a particular vessel. The manufacturing structure primarily facilitates the planning of steel construction for grand blocks and blocks of a ship, which

contain the items which are linked to production steps in the WBS. The organisational structure serves to coordinate cross-structural elements to form so-called production modules (collections), which are in turn pre-built components.

To emphasise the project-driven nature of ship and factory development, a product data model always refers to a corresponding main project. The project allows for organising development tasks, deliverables, milestones, phases and production activities, but is not within the primary focus of this work. It may encompass a range of sub-projects and tasks, all of which can be associated with product data. Further functionalities, such as open points and checklists, are also available. The incorporation of external project management applications is also feasible.

3.1. Hierarchical structures and corresponding views

As previously indicated, the development of a ship necessitates diverse perspectives, depending on the specific development domain and phase. These views require a corresponding hierarchy, mapped by the data model.

System structure: A vessel typically comprises multiple systems. Each system generally encompasses a significant quantity of elements, which can be hierarchically organised into system item groups. The relationships between the layers of a system can be described as "is part of" or "consists of". Based on Fig.1, specific system structures can be implemented. The SFI Group System, for instance, is widely adopted in the European shipbuilding industry.

Spatial structure: A spatial structure offers the possibility to represent the ship in a spatially structured way. Given the heterogeneity of spatial object types, such as compartments, rooms, decks or zones, a multiplicity of spatial structures may exist. Furthermore, hybrid structures are also implementable. Since a system usually extends over several spatial objects, system item groups or items can be assigned to multiple spatial objects. Relationships between the layers of a spatial structure can be interpreted as "includes" or "is located in".

Manufacturing Structure: The manufacturing structure is closely related to the spatial structures, but is specific to the planning of steel construction in shipbuilding, such as the construction of grand blocks and blocks. It is also used to plan the assembly of large items/components in the context of the blocks. Individual items and modules are assigned to the production steps by assigning them to blocks. This assignment is then used to generate block-specific component BOMs for production and procurement, as well as the building methodology for the assembly of the blocks and grand blocks. Relationships between the structural levels can also be described here as "includes" or "is located in".

Module Structure: The module structure enables the free organisation of information objects. In contrast to the structures mentioned above, these objects do not require structural purity. Elements from various structures can be aggregated and displayed through the hierarchical structuring of item collections. Freely structuring and linking objects is a practical necessity, for example, in the planning phase for the pre-assembly of main modules, such as externally assembled funnels, cabins, or mission equipment, where cross-structural linkages are essential. Using an item collection, the items from different systems are combined into a module and can be linked to spatial and manufacturing objects for further planning.

3.2. Items and related entities

An item is part of a system and can also be referred to as a functional location. It acts as a placeholder containing important meta information, such as requirements, a specific shipbuilding ID, such as an SFI Code, and location-specific metadata. E.g., the need for a pump component may be identified early in development, but its specific selection is deferred until later in the process. Accordingly, an item can refer to a supplier-specific component, which in turn can be selected from a component catalogue. This combination aims to increase the reuse of components, such as parts at different locations of a ship.

With document management being an integral part of ship design, items can be linked to documents, which have their own lifecycle and relationships, such as project tasks. Therefore, for example, the supplier contract for the procurement of a component for a functional location can be managed. Issues represent another key relationship, especially during ship construction. They document and track problems and improvements in construction, acceptance, and operation, enabling controlled resolution. Further item relationships are configurable per application.

4. Concrete Data Model Implementation

The presented data model was successfully implemented in the shipbuilding platform WAVE based on the Technology CONTACT Elements for both a partner and an external customer. At the system level, WAVE utilises the SFI system, which enables the clear identification of items and modules across disciplines and tools. For a better overview, compartments and rooms are also structured via decks and zones. Furthermore, the module structure is extended with BOM elements for the handling of prefabricated components with their respective foundations. Fig.2 illustrates the resulting data model, which consists of the view-specific object extensions and additional relationships.

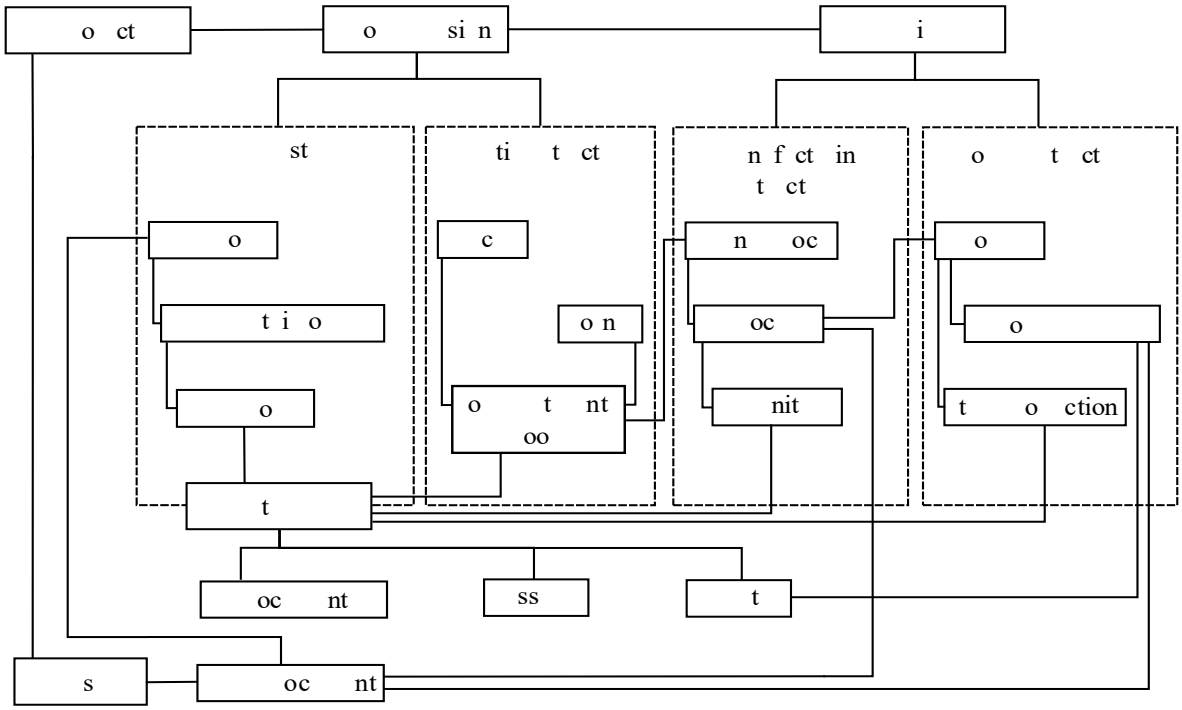


Fig.2: Concrete data model implementation in the WAVE shipbuilding platform

The implementation only includes elements and attributes from the Basic and Detail Design phase. More in-depth analysis already shows that milestones and delivery data from project management must be comprehensively linked with the elements to ensure smooth planning of the subsequent phases. For this purpose, CAD documents were supplemented with planning-relevant attributes, such as the delivery date, and integrated directly into the work breakdown structure as deliverables. Therefore, this approach can be utilised to plan and design externally fabricated modules efficiently. In addition to extracting block-specific BOMs, this also requires linking to module-specific 2D CAD drawings. The resulting elements are then used as functional components inside blocks and are connected to production steps within the building methodology for the grand block as part of the manufacturing process. With the following release versions, ship designs, full support for the SWBS, and the mapping of sister ships will be integrated into WAVE. Initial analyses have shown that the data model is only limited by missing attributes and elements, such as a ship type, which must be added and supplemented with PLM standard objects.

5. Discussion

The suggested multi-structure data model for a PLM system in shipbuilding allows for the creation of multiple functional and spatial structures to organise product data across all design phases and involved disciplines. A project structure, a fundamental element in shipbuilding, is not maintained in parallel with the product structure. Instead, it can be linked to individual data elements and is therefore interconnected with the key elements in the data model. This integrated methodology facilitates, for instance, document management, a critical aspect of shipbuilding. With the presented approach, Documents can be managed in the context of both product and project hierarchies.

Furthermore, CONTACT Elements, as the basis of the data model, features an open and modular platform. This design enables the system's adaptation to different application scenarios. A shipyard or design firm lacking prior experience with PLM systems can adopt a phased implementation approach. This approach involves initially transferring only partial processes or data, such as specific design phases. Conversely, the platform and data model enable integration of external expert tools, for example, simulations or CAD applications. The design processes can be progressively adapted for implementation within the PLM framework. Simultaneously, the PLM system can function as the Single Source of Truth from the beginning. Employee acceptance, specifically the initial recognition of added value, poses a significant challenge in PLM system implementation and digitalisation in general. For this reason, the data in the system should be labelled according to the terminology common in shipbuilding and training of the affected employee groups is of central importance. This problem constitutes an integral component of the SEUS project and is subject to further examination, as exemplified by the work of *Tacgin and Martinsuo (2025)*.

While the proposed data model presents notable advantages for shipbuilding applications, comprehensive validation remains outstanding. The solution presented herein represents an interim finding, requiring further research for completion. For this purpose, a functional validation will be carried out using both theoretical use cases and practical applications of European shipyards. Essential criteria are complete data integrity, ensuring the required interlinking of all information, and efficient information retrieval.

The data model maps shipbuilding's development phases, enabling linked, centralised information management. Although in the literature consulted, discrepancies with reality can be found in the course of the shipbuilding phases presented, the incorporation of additional data domains remains a plausible consideration. For example, *Aragão Fonseca et al. (2023)*, did not mention the basic design phase, although essential systemic foundations are developed within it. As reported by *Bronson et al. (2024)*, planning and process data domains are requisite in shipbuilding to effectively manage administrative and technical processes throughout the vessel's development and manufacturing phases. The same applies to the human domain, which includes the relevant people in development and production. Finally, the implementation of a Digital Twin in the context of a fleet management system, representing a comprehensive mapping of the ship's operational parameters, also justifies consideration. These developments may be part of further research with a focus on the interaction of data domains.

6. Summary and Outlook

This paper proposes a multi-structure data model for a shipbuilding PLM system. It integrates core spatial and functional elements for linking and organising product information. Adaptability to diverse development domains and phases simplifies the design of physically large and complex systems such as ships. In addition, these entities are interconnected with elements of project management, reflecting the project-driven nature inherent in ship development.

The data model, in conjunction with its target platform, CONTACT Elements, facilitates flexible adjustments to diverse circumstances. Since the introduction of a comprehensive PLM system may pose significant challenges, the model and associated platform offer the possibility of successive integration, which nevertheless already allows the provision of a single source of truth.

The current study represents an interim report, and comprehensive validation remains outstanding. A functional evaluation is planned based on theoretical and practical application scenarios derived from European shipyards. The implementation of the presented data model within SEUS has already yielded promising results. A final validation can only take place after the complete implementation across all phases. Central elements are the evaluation of data integrity and information retrieval. The integration of further information domains is to be considered. E.g., service planning and process information, and a human domain for administrative and technical processes, warrant further research. In addition, further questions arise from the interactions between existing and new domains, which can lead to entirely new research topics. All in all, multi-structure management provides the basis for making complexity in shipbuilding manageable and is therefore a promising approach for further research.

Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101096224.

References

ANDRADE, S.L.; MONTEIRO, T.G.; GASPAR, H.M. (2015), *Product Lifecycle Management In Ship Design: From Concept To Decommission In A Virtual Environment*, ECMS, pp.178–184

ANDREWS, D. (2013), *The True Nature of Ship Concept Design – And What it Means for the Future Development of CASD,* COMPIT Conf., Cortona, pp.33–50

ARAGÃO FONSECA, Í., FERRARI DE OLIVEIRA, F., GASPAR, H.M. (2023), *Open Framework for Digital Twin Ship Data: Case Studies on Handling of Multiple Taxonomies and Navigation Simulation*, Int. J. Maritime Eng. 165(A1), pp.23–42

BRONSON, J.A.; LUZ, F.H.P.; FONSECA, I.A.; GASPAR, H.M. (2024), *Graph databases for multi-domain taxonomies in maritime systems*, ECMS, pp.250–257

KAMOLA-CIESLIK, M. (2021), *Changes in the Global Shipbuilding Industry on the Examples of Selected States Worldwide in the 21st Century*, Eur. Research Studies J. XXIV(Issue 2B), pp.98-112

LEV Š K Ė , .; ULSTEIN, B.A.; GASPAR, H. (2017), *4GD Framework in Ship Design*, 16th COMPIT Conf., Cardiff

MISRA, S.C. (2015), *Design Principles of Ships and Marine Structures*, CRC Press

PAL, M. (2015), *Ship work breakdown structures through different ship lifecycle stages*, Int. Conf. Computer Applications in Shipbuilding, Bremen

PAPANIKOLAOU, A. (2010), *Holistic ship design optimisation*, Computer-Aided Design 42(11), pp.1028–1044

PAPANIKOLAOU, A.; BOULOUGOURIS, E.; ERIKSTAD, S.O.; HARRIES, S.; KANA, A.A. (2024), *Ship Design in the Era of Digital Transition: A State-of-the-Art Report*, IMDC Conf.

TACGIN, Z.; MARTINSUO, M. (2025), *An augmented reality solution for digitalisation training in shipbuilding: Systematic review and application development*, 19th INTED Conf., Valencia, pp.1997–2011

VAN DEN HAMER, P.; LEPOETER, K. (1996), *Managing design data: The five dimensions of CAD frameworks, configuration management, and product data management*, Proc. IEEE 84(1), pp.42–56

Solution Addressing Industry Concerns of Model-Based Approval

Juha Peippo, Rauma Marine Constructions, Rauma/Finland, juha.peippo@rmcfinland.fi
Jussi Puurula, Rapid Structural Design, Jyväskylä/Finland, jussi.puurula@rsd.fi
Michael Pudd, Rapid Structural Design, Turku/Finland, michael.pudd@rsd.fi
Dmitry Ponkratov, Siemens, London/England**,** dmitry.ponkratov@siemens.com

Abstract

Innovative shipbuilders, such as RMC, are looking for a unified design model for all design, approval and collaboration needs. As part of this work, together with Open Class 3D Exchange (OCX)-experts of RSD, we sought to streamline the ship design and approval process with Model-Based Approval (MBA). The main drivers for a successful shipyard are delivering high-quality products on time and within budget. Digital transformation, including model-based design and approval, has great potential to realize these industry drivers. There are natural concerns, whether MBA would require overly extensive modelling and expose too much product information. To address these concerns, we developed a concept in which an essential structural model is used as the design and approval basis, making the design process efficient while minimizing exposed information.

1. Introduction

"Commercial ship design and production are steadily under pressure to reduce time to market and overall production costs, as well as to improve product quality concerning multiple performance criteria that are to be met at the same time", *DNV (2023).* To stay ahead of the competition, innovative shipbuilders, such as RMC, are embracing new technologies and approaches.

In ongoing digital transformation, all stakeholders of the maritime industry focus on bringing their usual working routines into digital form. We can see that many shipyards develop their practices to cover the design and building process from the initial design to the production under a unified digital framework. Classification societies do the same and the maritime industry has high expectation of digital twins supporting operations. For successful digital transformation, reliable exchange of information commonly agreed between all stakeholders is a must.

Major ship designers and yards have realized the importance of using a 3D solution for the basic (class) design, *Pérez-Martinez and Pérez Fernández (2021)*, and drawings are becoming the product of 3D CAD models. Hence preparing drawings for approval is becoming an additional work-step, incurring additional cost and slowing the approval process.

In recent years, the OCX Consortium has developed OCX standard, which enables sharing a structural design model for approval, https://3docx.org/. Yet the shift is not merely technical. It touches trust, governance, and the way shipyards, designers, and class societies work together.

A class certificate is mandatory in shipping, and plan approval of a newbuild vessel is a key part of the certification process. In plan approval, class conducts a thorough evaluation of the main drawings, systems and installations to ensure that the design fully complies with international conventions, flag state requirements, and the classification rules, https://www.dnv.com/services/plan-approval-1577/ .

Structural plan approval drawings of today have become increasingly detailed to serve downstream production needs. Creating and sharing an equally detailed 3D model for approval would not only demand substantial additional effort but also expose sensitive production information. However, demonstrating structural compliance does not mandate a fully detailed production model. Most of the structures can be shown to be compliant from a model containing sufficient information for class to evaluate structural load carrying capability. Remaining details can be proven with a well-managed process and supporting documentation, in good collaboration with designer and class.

Without the requirement to submit approval drawings, shipyards could streamline the design process and share their design model with class, which can then autonomously extract needed information directly from the design model. This, following LEAN principles, would minimize waste and streamline the whole process.

In the following sections, we go through the ship design approval process and requirements to obtain a compliant structural design. We then present a concept showing how this can be achieved efficiently with a model-based approach

2. Ship Design and classification

The ship design process is a complex multi-variable optimization task with the primary goal of producing the production documentation necessary for manufacturing. During the design process, the functional requirements set for the product are transformed into plans that consider design and production technology, environmental impacts, and rule and regulation requirements.

2.1. Ship Design Phases

The ship design process is a complex, multi-stage procedure that transforms an initial concept into a fully operational vessel. This process is generally divided into three key stages: Concept Design, Basic Design, and Detailed Design, each with its specific goals and responsibilities.

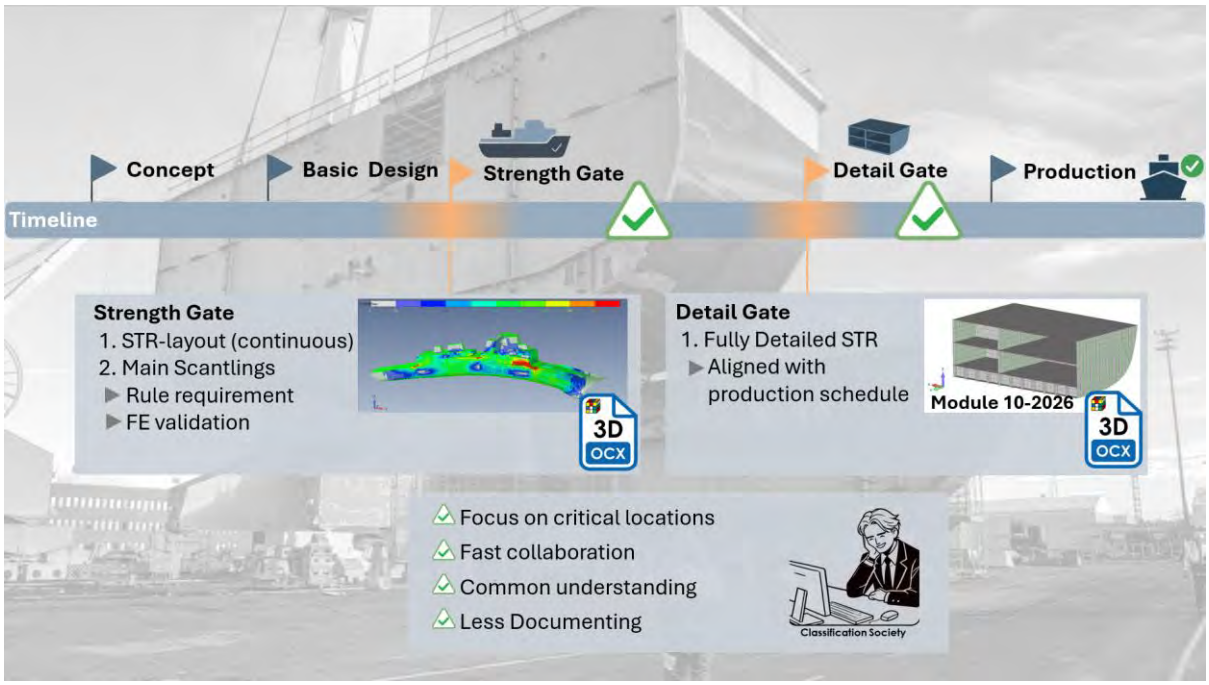


Fig.1: The main phases of the ship design process and conceptual gates for classification approval

The first phase, Concept Design**,** focuses on defining the ship's overall purpose and feasibility. Designers work closely with clients to understand the vessel's intended use, whether it is for cargo transport, passenger service, or military operations. Key factors like size, cargo capacity, speed, and environmental conditions are considered. Preliminary layouts are drawn, and initial calculations are made for stability, propulsion, and cost. This stage also involves evaluating potential technologies, materials, and design approaches, while ensuring compliance with safety regulations such as SOLAS (Safety of Life at Sea) and MARPOL (Marine Pollution). The primary goal here is to create a viable and functional concept that meets both operational and financial requirements.

In the Basic Design phase, the initial concept is refined and expanded. This stage involves developing a detailed ship layout, considering internal arrangements (e.g., engine room, cargo holds, accommoda-

tions) and external features. Designers perform more precise hydrodynamic calculations to optimize the ship's performance in water. The propulsion system is selected, and structural integrity is ensured through material specifications and strength calculations. Regulatory requirements are re-evaluated, and more accurate cost estimates are prepared. This phase sets the foundation for the final, detailed design.

In Detailed Design**,** a completely detailed structural model is prepared as the basis for production. Detailed features, such as openings, reinforcements, brackets, cutouts, notches, welds, and all production details are explicitly modeled. Creating, and especially updating, hundreds of thousands of details is laborious work, even with state-of-the-art tools. Hence, major structural design changes at this stage are unacceptable, as imposed cost and delays would have devastating financial impacts for the building shipyard.

For the shipyard it is crucial that each design phase has been completed successfully, with all stakeholders sharing a common understanding and assurance of the design before moving to next design phase. Ship design phases with conceptual approval gates are presented in Fig.1.

2.2. Ship classification process

Classification rules have evolved over decades, providing a framework to design and build safe and compliant vessels capable of withstanding the sea's often harsh and merciless conditions.

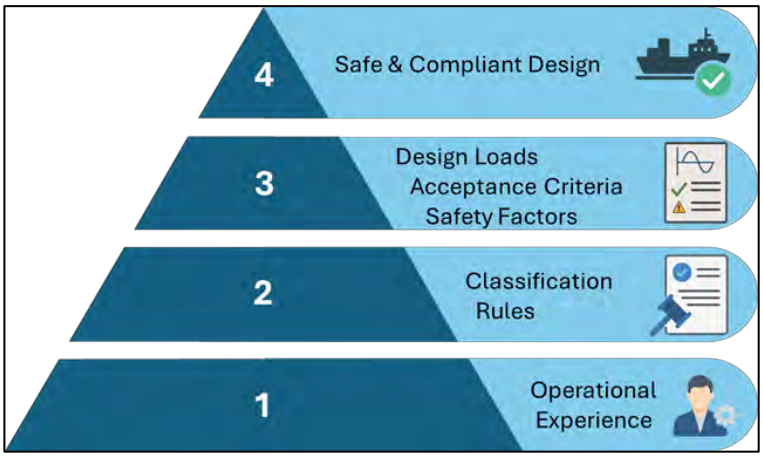


Fig.2: From operational experience to safe and compliant ship design

The design, construction, and operation of ships are subject to a rigorous classification process that serves to ensure the safety, reliability, and environmental responsibility of maritime vessels. The ship classification process is a multi-layered, iterative process. Its primary objective is to verify that every vessel conforms to the technical standards and structural integrity required by both international conventions and the specific rules set forth by the chosen classification society.

The classification process begins in the early stages of design. As soon as the principal parameters of the vessel, such as hull configuration, size, and the arrangement of major systems, are defined, the design team initiates contact with classification representatives. This proactive engagement is crucial: it allows for systematic review and iterative feedback on preliminary plans, ensuring that every aspect of the vessel is considered in light of prevailing standards.

During basic design, the design team prepares comprehensive documentation, including plan approval drawings with supporting documents and calculations. Structural layouts, scantlings and details are presented in the drawings, which are submitted for plan approval. Supporting calculations, validating rule compliance, are typically submitted as information.

In plan approval, class conducts a thorough evaluation of all the submitted drawings, to ensure the design fully complies with classification rules and requirements of international and flag state. Assurance of approval, without major changes, is essential for the shipyard, before the block fabrication begins.

Lastly, a newbuilding survey is conducted. In the block survey, the class inspector checks that structures are built according to the approved drawings. At this stage, any new findings requiring change would have a devastating impact for building cost and schedule.

2.3. Information Exchange

The current classification process is still mainly 2D document-based, with a large number of drawings submitted for plan approval. Although most shipyards now design and construct using 3D models, exchanges with classification societies continue to rely on 2D deliverables. This has critical implications, *BV (2019):*

- The designer extracts and prepares an suite of 2D drawings for classification
- Classification societies often need to rebuild a digital verification model from these 2D documents in order to run their checks
- Comments returned on the drawings then have to be re-applied to the 3D model and across multiple documents

This creates an iterative, time-consuming, and error-prone loop, *Astrup and Cabos (2017).*

A more effective approach is digital, model-to-model exchange. The Open Class eXchange (OCX) format has been developed to capture the needs of both classification societies and shipbuilders for fully digital information transfer, *DNV (2023)*. OCX can transfer structural information in the necessary basic design context, *Gusani et al. (2023).*

Given that most designers already create rich 3D models, the logical next step is to extract approval-relevant information directly from those models and conduct reviews digitally, rather than through manual 2D interpretation. Moving from paper-like drawings to 3D, data-centric design and approval enables, *DNV (2023), Astrup and Cabos (2017):*

- a single source of truth, where the product model governs the approval set
- automation of rule checks and pre-checks, reducing the number of iterations
- faster change management, with precise, object-level design changes instead of reissuing entire drawing packs
- reduced re-work and inconsistency risk

The industry's 2D-centric exchange imposes avoidable latency and quality risks on both shipyards and classification societies. Model-based exchange (via OCX) provides the foundation for a streamlined, collaborative approval process.

3. Structural Strength Design

The most fundamental structural requirement is that a ship's structure must safely withstand all relevant load scenarios. Technically, this is expressed as:

$$F_{rd} > F_{design} * SF$$

F_{rd} is the structural resistance, F_{design} the applied design load, and *SF* the safety factor. This simple inequality captures the essence of structural safety: resistance must exceed the load response, with margins that balance safety and economy.

Classification rules provide practical methods to apply this requirement in design. Rules define the relevant design loads and combinations, acceptance criteria, and safety factors, drawing on decades of operational experience. By following these principles, designers ensure safe and compliant structures.

3.1. Workflow

A structural system is only as strong as its weakest link. A top-down design approach is typically applied to ensure a continuous and robust chain:

1. Continuous structural layout – uninterrupted load paths
2. Compliant structural scantlings – structural members with sufficient strength
3. Good structural details – local continuity and avoidance of hard points

The global arrangement of main strength members is decisive, as it sets the foundation for the entire structure. Once established, primary structural members and secondary stiffener arrangements follow logically, with initial scantlings obtained from prescriptive calculations. Load distribution of initial design is then simulated with FE-analyses, which usually results in some design changes and reinforcements. Resulting in a good structural design, with continuous load bearing structures capable of carrying the required loads.

Structural system is only as strong as its weakest link and using well-established standard details throughout the vessel also remains essential.

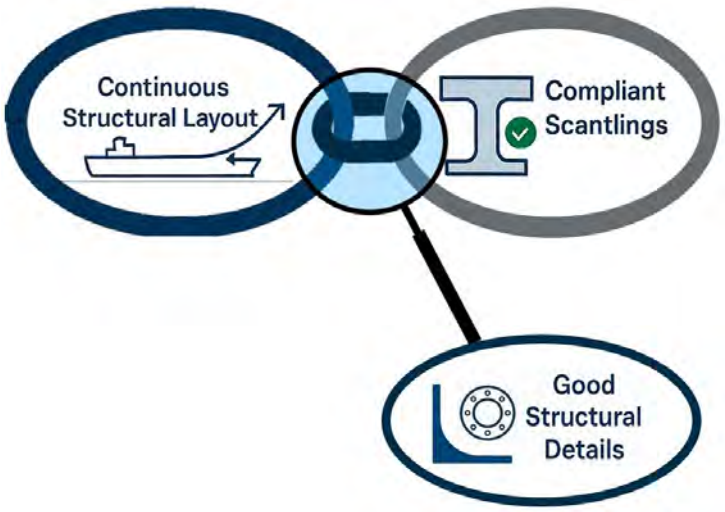


Fig.3: Strong structural chain

3.2 Design Documentation

In a conventional drawing-based approach, a large package of 2D drawings, documents and reports are prepared and maintained for plan approval. Such documentation involves considerable effort to prepare and maintain. Change management becomes especially cumbersome, when information has dispersed across multiple documents.

With modern tools, creating a lightweight 3D model, which captures structural layout and scantlings sufficient for basic design structural assessment, is feasible and efficient. Our experience indicates that preparing such a model requires an effort comparable to producing conventional plan-approval drawings.

Having a unified design model would minimize rework and enable automated workflows. Instead of maintaining and interpreting documents manually, the designer’s effort would contribute to improving

the design rather than maintaining documentation As an practical example, unified design model could be used as a single-source-of-truth for automated calculations and FE modelling, substantially reducing manual design effort.

3.3. Strength Calculations

Rules typically define the governing load scenarios and combinations for both global hull-girder loads and local operational loads. In practice, many members contribute to both systems. For example, inner-bottom plating carries global axial stresses from hull-girder bending while at the same time being subjected to cargo and/or tank pressures. Therefore, load responses must be evaluated for the combined effects of global and local loads, with the most severe combination determining the governing strength and buckling checks.

Structural calculations yield responses (deformations, forces, stresses) for all relevant load combinations, which are then checked against rule-based strength criteria. Typical criteria include:

- Minimum scantlings, ensuring a baseline of compliant structures
- Deformation limits, avoiding slender or overly flexible arrangements
- Stress limits, maintaining utilization below yield criteria
- Buckling criteria, preventing instability under compression

All structural members, under all applicable load combinations, must satisfy these criteria.

Prescriptive rule calculations are effective for obtaining minimum scantlings for plates and stiffeners. For hull girders and conventional arrangements, such calculations are often sufficient. However, for novel designs they may require conservative assumptions, which can lead to heavier structures and even discourage novel solutions.

Advanced simulations, particularly finite element analysis (FEA), can accurately capture how loads are distributed among structural members. Most rule frameworks already require global FEA and, for critical areas, local FEA as part of plan approval. Nevertheless, FE modelling is often considered resource-intensive and is therefore under-utilized when not explicitly mandated.

Having a unified design model as the basis would enable automated workflows, making more extensive FE-calculations feasible. With more precise responses, designers can meet rule criteria without relying on excessive conservatism, enabling optimized and innovative solutions.

4. Case Study: Solution for Initial Basic Design

RMC is looking for a unified design model for all design, approval and collaboration needs. This unified design model would be single-source-of-truth for

- Structural design
- Prescriptive calculations
- FE-calculations
- Design collaboration with all disciplines
- Approval collaboration.

We started implementation work from initial design, aiming to submit the first approval package to plan approval as 3D OCX. This design phase seemed most feasible to implement, since calculation models are mandatory whereas drawings are not. For testing and demonstration, we created a fictitious design of a 135-m naval destroyer. Although not intended for production, this design captures the structural complexity and approval challenges typical of modern naval vessels.

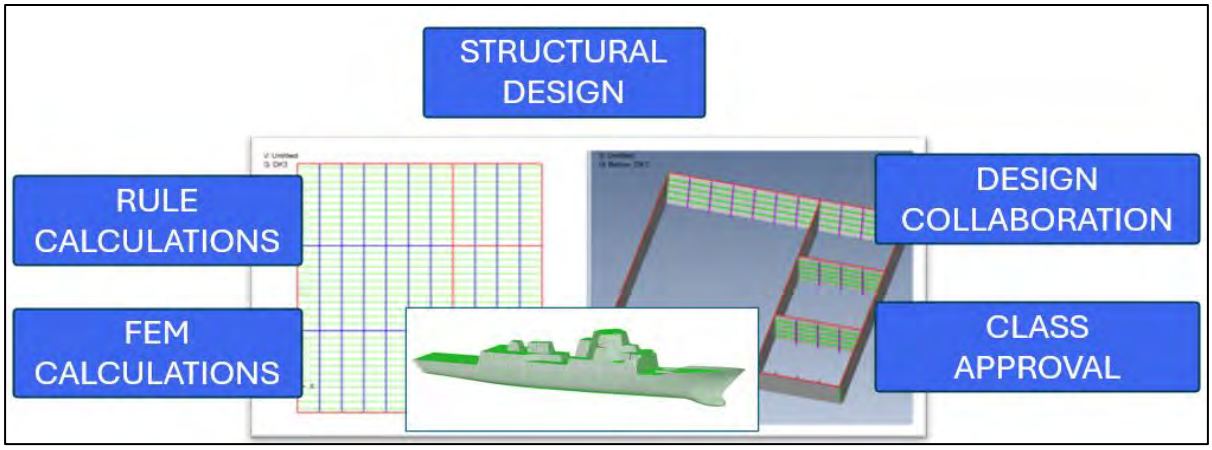


Fig.4: Unified design model

4.1. Structural Design

First, we search for feasible structural design, where structures are generally in correct locations and scantlings are reasonable.

Drawing the structural layout and defining exterior surfaces is considered typical design work. After these preparations the 3D model generation effort was very minimal, as RSD developed a prototype, that generates 3D OCX-panels for typical ship structures from the smart structural layout. Resulting structural model is presented on following figure.

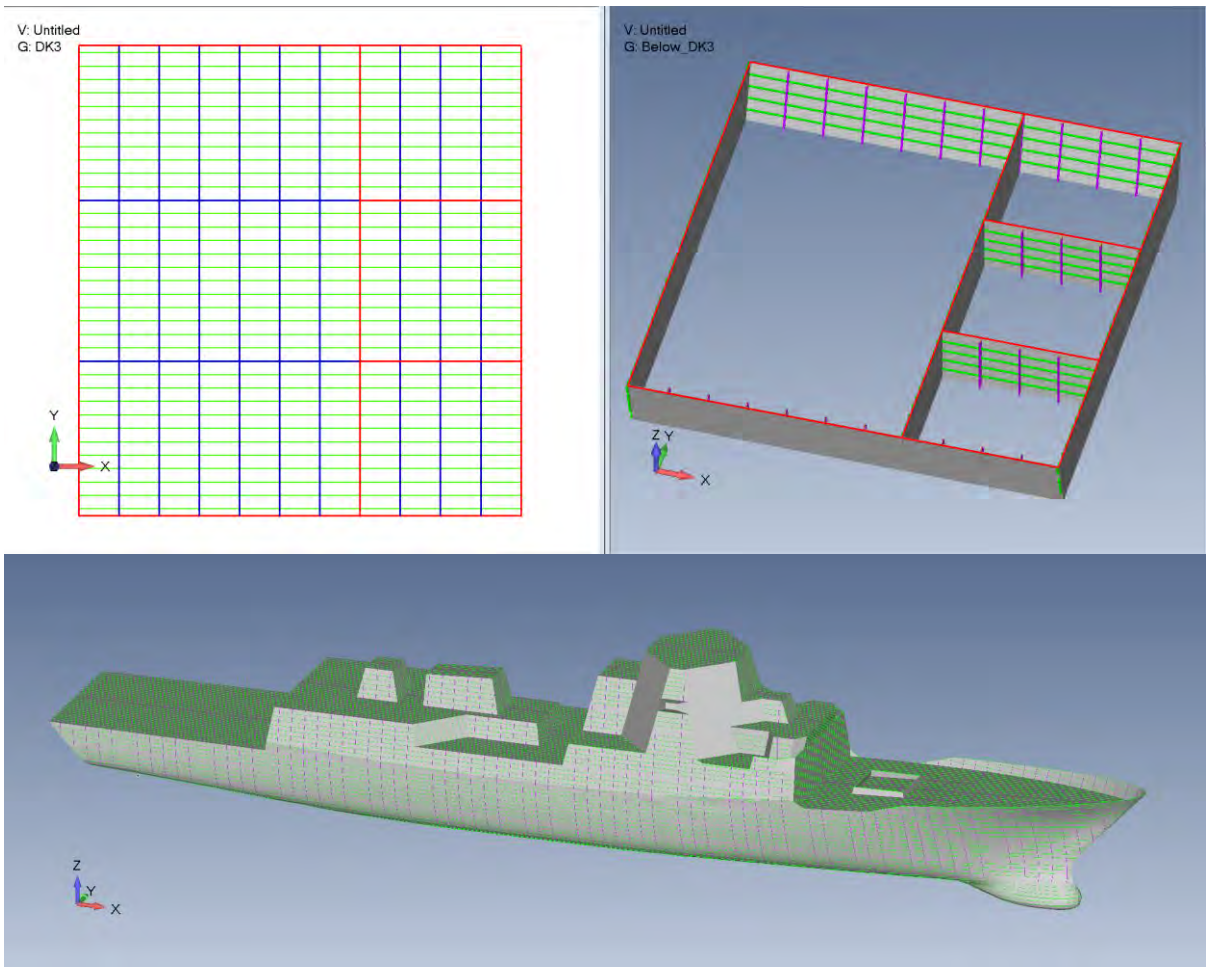


Fig.5: Smart structural layout and the resulting 3D model

4.2. Strength Calculations

Our initial unified design model is natively OCX and it presents plates and stiffeners with correct locations, scantlings and spans. Hence it can be used as basis for rule and FE-calculations.

For prescriptive calculations, designers can use classification society tools supporting OCX, such as DNV's Nauticus Hull, which we used to test re-using our OCX-model. DNV provides guidance to run a prescriptive cross-section calculations based on the imported OCX-model. OCX contains all the information to fully reconstruct an calculation model. For details, see *DNV (2025b)*.

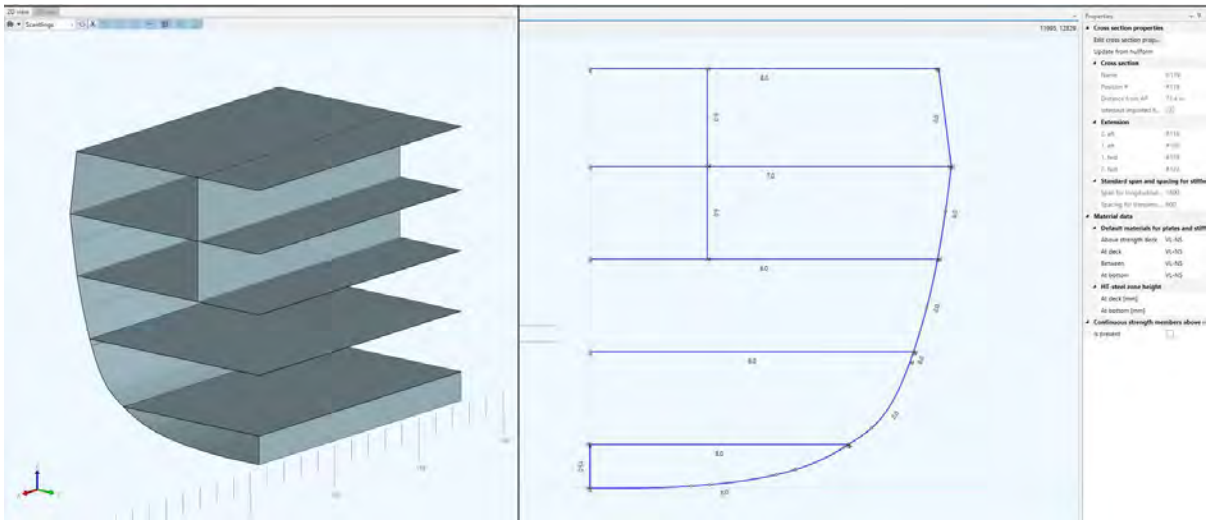


Fig.6: Midship section OCX-panels in DNV Nauticus Hull

For FE-calculations, most time-consuming work is modelling. This manual effort was replaced by automated meshing using unified data model as the basis. RSD's meshing prototype produced 95% calculation ready mesh, requiring only minimal mesh edits. Classical loading and stress assessment for non-CSR vessel was streamlined and performed with RSD Add-On for Simcenter FEMAP.

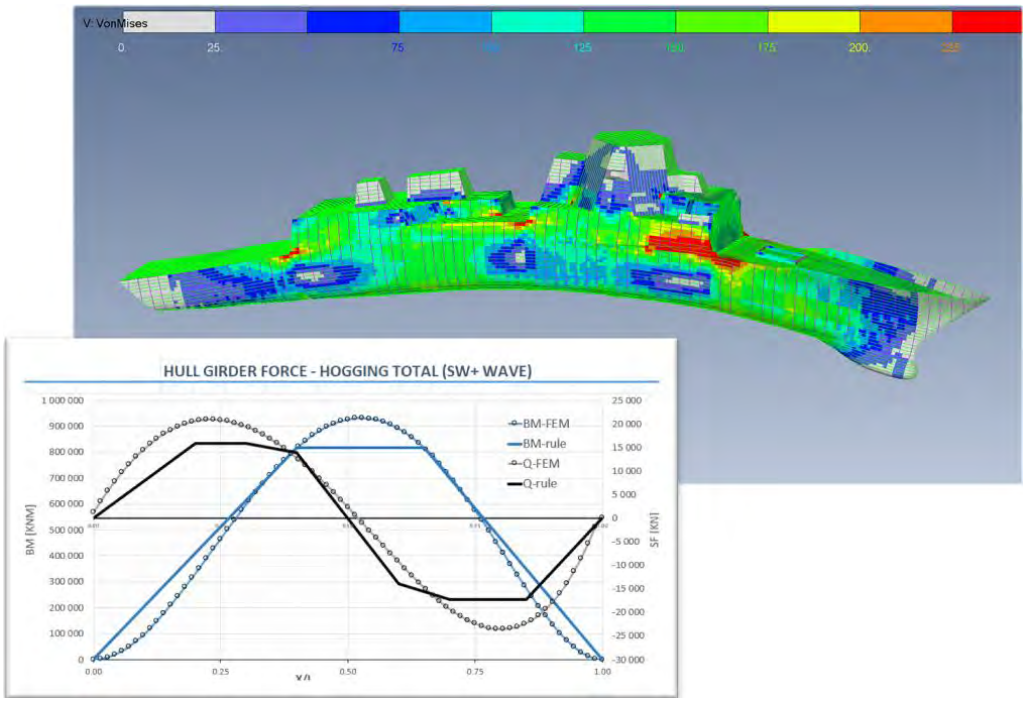


Fig.7: Global FE-results

4.3. Approval

We have accomplished a structurally compliant and safe global design, with continuous structural layout and compliant scantlings. As the design itself is an OCX-model, we can submit it to class for early involvement, without any additional work. Class can then extract the necessary approval information directly from the model and perform their review digitally, *DNV (2023).*

At ICCAS 2024, RSD proposed a novel approach to replace paper-like PDF reports with FE models organized as OCX. This approach reduces the reporting effort for designers and enables further automation in class review, *Puurula et al. (2024).*

After class review, we have obtained concrete assurance that our design is compliant at global level. Hence, we can continue to evolve our design with confidence. As class now has holistic shared view of our design, we expect clear and accurate information requests from class. With good collaboration, both class and designers focus on improving the design, not the documentation.

5. Discussion

The digital transformation provides great opportunities for all stakeholders of the entire maritime industry. The change is not merely technical. It touches on trust, governance, and the way shipyards, designers, and classification societies work together. There are many natural concerns and challenges that must be addressed together. We hope that our article written from a ship designer's point-of-view will contribute to the valuable work that OCX consortium and entire maritime industry are doing.

Model-based collaboration benefits from the single-source-of-truth principle, eliminating media breaks, duplicated data entry, data quality issues, and wasted effort, *DNV (2023).* Compared to relying on several separate documents, the MBA workflow enables efficient and transparent collaboration. Changes and review discussions are attached directly to the design context, ensuring clear traceability and a continuous digital thread. With MBA, review threads are no longer dispersed across multiple PDF documents but remain directly connected to the evolving design model.

Open Class 3D Exchange (OCX) schema provides a great standardized way to present ship structures with information needed for class approval. However, a wide variety of OCX-implementations exists, as different CAD-systems and users have varying modelling approaches. Which makes it challenging for consuming applications to extract the information needed to build native calculation models.

Perfecting interoperability of OCX-exchanges is crucially important, which OCX-implementor forum (OCX-IF) is addressing. However, as shipbuilders carrying the risk of incurred costs of delays, we cannot fully rely on the promise of 100% interoperability for critical submissions as the only option. For risk mitigation, we expect receiving classification applications and procedures to have flexibility to overcome issues arising from nearly perfect exchanges.

In our case study, within a week we modeled, calculated and submitted a conceptual design. This shows model-centric working to be highly efficient and, combined with the clear benefits of model-based collaboration, makes the MBA concept very compelling for shipbuilders.

The model-based approval process is not yet commonplace. Implementing this approach still requires commitment and development from all parties before the full benefits of the new operating model can be realized. However, it is easy to see that the model-based approval process provides a catalyst that will positively impact various stages of the ship delivery process.

References

ASTRUP, O.; CABOS, C. (2017), *A model based approval process for basic hull design*, ICCAS Conf., Singapore, pp. 107–116

BV (2019), *Digital Classification*, BV Technology Report #06, Bureau Veritas, Neuilly-sur-Seine

DNV (2023), *Paving the way for 3D model-based class approval*, White paper, DNV, Hovik, https://www.dnv.com/services/dnv-3d-approval-platform-248148/

DNV (2025), *Import of OCX models [PDF]*, Nauticus Hull User Manuals 20.31, DNV, Hovik

GUSANI, S., RADONIC, M.; PUURULA, J. (2023), *OCX Standard and Structural Model Reuse in the Shipbuilding Design*, 22nd COMPIT Conf., Drübeck, pp.391–406

PÉREZ-MARTINEZ, J.; PÉREZ FERNÁNDEZ, R. (2021), *Shipbuilding 3D CAD Tools as an Integrated Solution from Concept to Product*, Ship Science & Technology 15(29), pp.45–57

PUURULA, J.; PUDOVKIN, M.; GUSANI, S. (2024), *Model-Based FEA Reporting for Streamlining Approval Workflows*, ICCAS Conf., Genova

Investigating Minimum Sufficiency of Data and Feasibility of Peripheral Sensors for RUL Prediction for a Lab-Scale Ship Machinery Plant

Christian H. Manohar, University of Michigan, Ann Arbor/USA, chmano@umich.edu
Alexander D. Manohar, University of Michigan, Ann Arbor/USA, alexmano@umich.edu
Connor W. Arrigan, University of Michigan, Ann Arbor/USA, arriganc@umich.edu
Christopher J. De Martinis, University of Michigan, Ann Arbor/USA, cjdemart@umich.edu
David J. Singer, University of Michigan, Ann Arbor/USA, djsinger@umich.edu

Abstract

Accurate prediction of ship systems' Remaining Useful Life (RUL) is essential for maritime safety, reliability, and cost-effectiveness. Hardware-based failure data for RUL research is scarce, and to address this issue, Dr. Stephen A. Olson developed a lab-scale electric ship machinery plant at the University of Michigan, featuring various systems and real-time control. We previously explored the effectiveness of current RUL techniques like Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNNs) on Dr. Olson's plant's cooling and fueling systems. This study aims to evaluate these same techniques on the plant's electrical, propulsion, and mission systems and explore which of the plant's data sources are most vital for RUL method development.

1. Introduction

In modern maritime operations, predictive maintenance has become a vital component, crucial in accurately estimating the Remaining Useful Life (RUL) of ship parts. This approach helps enhance safety, cut costs, and boost operational efficiency. Historically, maintenance strategies such as reactive repairs and scheduled checks often led to unexpected downtimes or unnecessary part replacements, creating significant financial and logistical issues. Advances in sensor technology and data analytics have revolutionized maintenance practices, paving the way for condition-based approaches that offer more accurate forecasts of when components will degrade or fail. This transition is especially important for valuable equipment like azimuth thrusters, turbochargers, and diesel generators—assets that operate under tough environmental conditions and complex loads, *Velasco-Gallego et al. (2023); Kongsberg (2025a,b).*

Although progress has been made, accurately predicting remaining useful life (RUL) continues to be difficult because of the complexity and diversity of shipboard machinery. Many ships use manufacturer-specific monitoring systems that isolate sensor data, hindering comprehensive analysis across different systems and forcing diagnostics to focus only on individual components. Additional hurdles include limited satellite bandwidth, data loss from sensors, and irregular sampling rates, all of which hinder the ability to develop a comprehensive view of the vessel's health.

This study continues from the authors' previous work, examining the efficacy of the Marine Engineering Laboratory (MEL) at the University of Michigan, which was developed by Dr. Stephen A. Olson and Professor Timothy McCoy, *Olson et al. (2024).* The MEL is a lab-scale electric ship machinery plant that can trigger failures on command, such as clogs and leaks. We previously examined the efficacy of several state-of-the-art machine learning-based RUL models on the MEL to explore the lab's potential as a platform for developing RUL models, *Manohar et al. (2025)*. We examined the fueling and cooling systems, as was initially examined in *Olson (2024).* In this study, we expand our analysis of the efficacy of the MEL to the other systems, including the electrical, propulsion, and mission systems, to determine their usefulness for RUL models and their potential efficacy in future model development. In particular, we utilize the same models from our prior work, which are two Long Short-Term Memory (LSTM) models, one created by Dr. Olson in *Olson (2024)* and another created by *Yang et al. (2022),* and a Convolutional Neural Network (CNN) and LSTM model developed by *Li et al. (2019)*. We omitted the CNN-LSTM-Autoencoder model developed by *Ren et al. (2021)* due to the results from our prior work.

By extending our previous work to capture the other systems within the MEL, we can examine the plant's total efficacy and provide general guidance for how the MEL can be used in future model development.

2. Background

The following sections will detail the background necessary for this paper. We will describe the Marine Engineering Laboratory (MEL), machine learning concepts, the data used for this study, and its remaining useful life.

2.1. Marine Engineering Laboratory Test Bed

The Marine Engineering Laboratory was designed and developed by Dr. Stephen A. Olson and Prof. Timothy McCoy at the University of Michigan's Department of Naval Architecture and Marine Engineering *Olson et al. (2024)*. The laboratory emulates shipboard machinery systems by combining six coupled systems. This paper examines three of these systems: the propulsion, mission, and control systems. For the fueling and cooling system descriptions, refer to *Manohar et al. (2025)*, and *Olson (2024)*.

2.1.1. Electrical System

The electrical system emulates a commercial architecture for an all-electric ship. It includes three 7.5 kW 3-phase generator sets powered by Variable Frequency Drive (VFD) fed induction machines, which can emulate gas and diesel engine transient behavior. Additionally, it has two high-voltage, 3-phase switchboards at 280 V. The electrical system also has 9,600 kWh of energy storage and feeds into the electrical propulsion system, which is described in the next section. The one-line diagram of the electrical system can be found below in Fig.1.

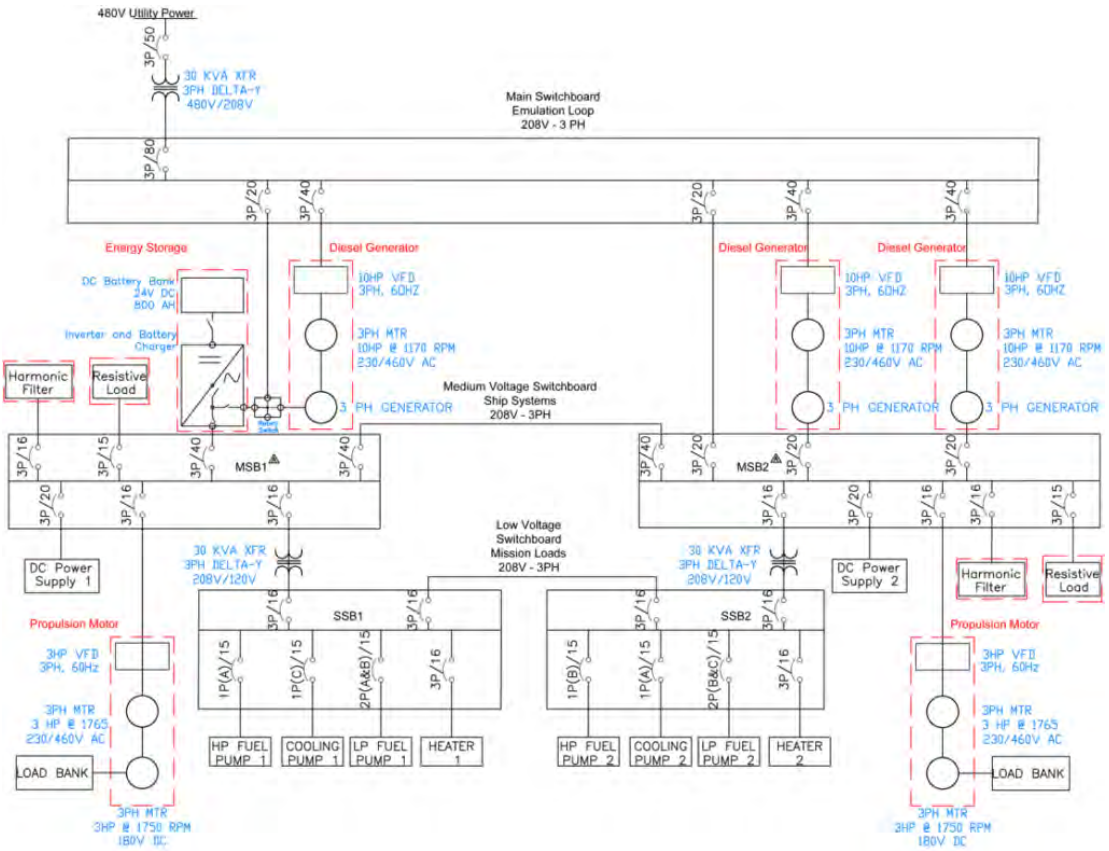


Fig. 1: One-line diagram of the electrical system.

The electrical system sensors can be partitioned into three sets: one for electrical management, a set related to energy generation, and one for energy storage. The electrical management set contains 20 sensors that provide information about the state of the overall platform. These are found in Fig.2. Eight of these 20 sensors are related to Input/Output (I/O) operations and are not directly applicable to the RUL task. Four of the remaining 12 sensors correspond to the measured input from the local electrical utility, which is used to help power the system. Consequently, these sensors are also not relevant to the RUL task. Therefore, there is a total of eight sensors of the electrical management sensor set that are useful for the RUL task. Fig.2 shows the 3-phase voltage sensors for the main switchboards (MSB1 and MSB2), and the sub-switchboards (SSB1 and SSB2). These sensors are directly related to the functioning of the electrical system and the platform. Therefore, they are relevant to the models, and the models employ them to identify the RUL of the platform.

Component	Signal	Units
Utility Input	3 Phase (A,B,C) - Voltage	Vac RMS
Utility Input	3 Phase (A,B,C) - Amps	A RMS
Utility Input	Neutral	A RMS
Utility Input	Ground	A RMS
MSB1	3 Phase (A,B,C) - Voltage	Vac RMS
MSB1	3 Phase (A,B,C) - Voltage	Vac waveform
MSB2	3 Phase (A,B,C) - Voltage	Vac RMS
MSB2	3 Phase (A,B,C) - Voltage	Vac waveform
SSB1	3 Phase (A,B,C) - Voltage	Vac RMS
SSB1	3 Phase (A,B,C) - Voltage	Vac waveform
SSB2	3 Phase (A,B,C) - Voltage	Vac RMS
SSB2	3 Phase (A,B,C) - Voltage	Vac waveform
Electrical Relay	SSB1 to SSB2	I/O
Electrical Relay	Shore Tie	I/O
Electrical Relay	Main Input	I/O
Electrical Relay	Resistive Load 1	I/O
Electrical Relay	Resistive Load 2	I/O
Electrical Relay	MSB1 to SSB1	I/O
Electrical Relay	MSB2 to SSB2	I/O
Electrical Relay	MSB1 to MSB2	I/O

Fig.2: Electrical system signal list

Component	Signal	Units
3 Phase Alternator	3 Phase (A,B,C) - Voltage	Vac RMS
3 Phase Alternator	3 Phase (A,B,C) - Amps	A RMS
3 Phase Alternator	3 Phase (A,B,C) - Voltage	Vac waveform
3 Phase Alternator	3 Phase (A,B,C) - Amps	A waveform
VFD	Load Torque Applied	% of Max Torque
VFD	Start / Stop	I/O
Relay	Gen1 to Bus	I/O
AVR	Voltage Setpoint	Vac RMS

Fig.3: Emulated generator set signal summary

The electrical system has three generators, each with eight sensors, Fig.3. Two sensors are related to I/O and irrelevant to the RUL task. However, the remaining six sensors, measuring the voltage and amperage, are relevant and are employed by the models for RUL. Sensors from all three generators are utilized in the models.

Finally, the electrical storage has eight sensors related to the battery bank and inverters, Fig.4. One of the sensors is for I/O and is not used. The remaining seven sensors measure the battery's current and temperature, the battery bank's voltage, and the voltage and amperage of the inverters.

Component	Signal	Units
Battery Bank	DC Current	A DC
Battery Bank	Temperature	Deg C
Battery Bank	Voltage	Vdc
Inverter	3 Phase (A,B,C) - Voltage	Vac RMS
Inverter	3 Phase (A,B,C) - Amps	A RMS
Inverter	3 Phase (A,B,C) - Voltage	Vac waveform
Inverter	3 Phase (A,B,C) - Amps	A waveform
Inverter	Relay Inverter to Bus	I/O

Fig.4: Energy storage signal summary

2.1.2. Propulsion System

The propulsion system contains two 2 kW VFD-fed induction motors that power excited DC motors, which emulate the propeller load. The generated power is dissipated in resistors. There are nine sensors within the two propulsion systems, Fig.5. Four of these sensors are related to I/O operations and are not used by the models. The remaining sensors measure the current and speed of the VFDs, the current and voltage of the DC motors, and the voltage command of the field power supply. These are potentially relevant to RUL identification and are employed by the models.

Component	Signal	Units
VFD	Phase A - Current	A RMS
VFD	Speed Command	Hz
VFD	Relay	I/o
VFD	Enable forward	I/o
VFD	Enable rev	I/o
DC Machine	Load Current	A DC
DC Machine	Load Voltage	V DC
Field Power Supply	Relay	I/o
Field Power Supply	Voltage Command	V DC

Fig.5: Propulsion system signals

2.1.3. Mission System

The mission system emulates mission loads via two programmable load banks, each fed by controllable DC power supplies from the main switchboards. These mission loads can provide up to 7.5 kW of load on the switchboards and simulate dynamic loads, including an electromagnetic railgun, an electromagnetic aircraft launching system, and a laser weapon system. However, these loads are not employed for this work. A 2 kW linear resistive load is attached to the main

switchboards. Each load bank contains four sensors, as found in Fig.6, one related to I/O and not used in the models. The remaining three sensors measure the voltage and current of the DC power supply and the current of the DC load placed by the load banks. Consequently, these sensors are employed by the models.

Component	Signal	Units
DC Power Supply	Voltage setpoint	A DC
DC Power Supply	Current Limit	A DC
DC Load	Program Trigger	I/O
DC Load	Current Command	A DC

Fig.6: Mission system signals

2.2. Remaining Useful Life

Remaining Useful Life (RUL) refers to the projected duration a machine or part can operate efficiently before maintenance or replacement is needed. This metric is crucial in prognostics and health management (PHM), facilitating predictive maintenance that enhances operational efficiency and reduces costs. RUL forecasting utilizes data from monitoring conditions, degradation models, and algorithms to predict machinery health and potential failure timelines, *Baru et al. (2023), Das et al. (2010)*.

Methods such as vibration analysis and artificial intelligence are employed for rotational machinery to identify anomalies and forecast degradation patterns. These techniques are essential in dynamic environments where operational conditions fluctuate, *Huang et al. (2024), Mulay et al. (2022)*, *Zhang et al. (2021), Nair et al. (2019)*.

In maritime settings, machinery faces unique challenges, requiring effective maintenance strategies to ensure safety and reliability. RUL estimation aids shipboard reliability programs by providing insights into the condition of critical components. Leveraging real-time sensor data and machine learning allows ship operators to detect early signs of wear and schedule proactive maintenance, ultimately lowering operational expenses.

Recent advancements in RUL prediction, including deep learning methods and hybrid maintenance strategies, enhance accuracy and adaptability in maritime operations. However, data variability and the need for robust models persist. Emerging technologies like artificial intelligence and the Industrial Internet of Things (IIoT) offer solutions to improve prediction accuracy and optimize machinery performance, *Li et al. (2024)*.

2.3. Machine Learning

Machine learning has become a highly prevalent field that aims to build effective computer models to perform various tasks through a defined learning process. The most applicable tasks models can perform for this study’s purposes are classifying failure profiles and identifying systems' RUL. Over time, different model architectures, mostly different forms of neural networks, have been developed and widely used. The different models utilized in this study involve Long-Short-Term Memory cells (LSTMs) and Convolutional Neural Networks (CNNs).

2.3.1. Neural Networks

Generally, neural networks use a series of linear transformations to approximate functions, *Abdi et al. (1999)*. The weights of the linear transformations are learned by leveraging labeled data and back-propagating their gradients throughout linear transformation layers. Therefore, neural networks can approximate complex functions and are employed in various complex and difficult nonlinear tasks.

2.3.2. Long Short-Term Memory (LSTM)

LSTM is a type of recurrent neural network that aims to capture long-term dependencies in data, *Hochreiter et al. (1997)*. Frequently, neural network architectures have issues with maintaining long-term relationships in data, and LSTM is designed to hold positional and relational information about data when a model is learning. It achieves this by being composed of a single cell and three gates: an input gate, an output gate, and a forget gate. The input and output gates control the flow of information, and the forget gate contains information about positional relationships between data. LSTMs are used in many tasks and can be very useful for classification.

2.3.4. Convolutional Neural Networks (CNNs)

CNNs are a type of neural network that involves convolving a filter (or kernel) over data to capture spatial relationships that may be present, *LeCun et al. (1989)*. They are a very popular type of model, often being used in tandem with other architectures like LSTM to create more complex and flexible models.

3. Methodology

The following section details the ascribed RUL models and the methods utilized in the case study. Specific model architectures are described fully in *Manohar et al. (2025)* and thus will not be covered exhaustively here. All models were recreated using PyTorch, a widely used deep learning library in Python, *Paszke et al. (2019)*.

3.1. Models

We utilize the Base-LSTM, LSTM, and LSTM-CNN machine learning models, the same as in our prior work, *Manohar et al. (2025).* For brevity's sake, they are not repeated here. Details can be found there. The hyperparameters of the models will be included here for reference.

3.2. Hyperparameters

For Dr. Olson's Base LSTM model, the hyperparameters in Table I are used. For the LSTM-based model by *Yang et al.,* the hyperparameters in Table II are used. For the LSTMCNN model by *Li et al.,* the hyperparameters in Table III are used.

Table I: Base LSTM Model Hyperparameters

Parameter	Value
Learning Rate	0.005
Number of Epochs	500
Optimizer	Adam
Loss Function	Root-Mean-Square Error
Model Selection	Best Validation Loss

Table II: LSTM Model Hyperparameters

Parameter	Value
Learning Rate	0.005
Number of Epochs	750
Optimizer	Adam
Loss Function	Root-Mean-Square Error
Model Selection	Best Validation Loss

Table III: LSTM-CNN Hyperparameters

Parameter	Value
Learning Rate	0.005
Number of Epochs	500
Optimizer	Adam
Loss Function	Root-Mean-Square Error
Model Selection	Best Validation Loss

The Root-Mean-Square Error loss function is defined by:

$$RMSE = \sqrt{\sum_{y=1}^{n} (y_i - \hat{y}_i)^2}$$

y_i is the true label of sequence i and $\hat{y}_i$ is the model's predicted label for sequence i, *Hodson (2022).* During training, the best model was selected using the model that recorded the minimal loss on the validation set.

The Adaptive Moment Estimation (Adam) optimizer *Kingma (2015)* is used as the optimizer for each model.

3.3. Data Preprocessing

Several stages of data preprocessing are used to test the efficacy of these various model architectures on the Marine Engineering Lab Test Bed. To see the preprocessing stages, please refer to our prior paper, *Manohar et al. (2025)*.

3.4. Model Training and Testing Procedures

The data were partitioned into training, testing, and validation sets to train and test the model architectures. The training and validation sets are used during the training process, and the testing set is used to verify the models' effectiveness after training has finished. The split used, referred to as "Partition A" throughout the paper, delegated 80% of the data to training, 10% to validation, and the remaining 10% for testing.

3.5. Model Retraining and Retesting

After training and testing the model architectures on the electrical, mission, and propulsion systems, we observe and note their efficacy. If sufficiently effective, the models are then retrained on new data sets comprised of one of the original sensor groups, i.e., sensor group 1 ("s1_g1"), and any of the new sensor groups for which the models proved effective. Once retraining is complete, these new models are tested on the latest data sets formed from the original and new sensor groups.

4. Results

The following charts are presented with the following features. All three models are detailed on each chart in order of Base, LSTM, and LSTMCNN in order of *Olson (2024), Yang et al. (2022),* and *Li et al. (2019),* respectively. For each presented model, the data corresponding to the electrical sensor group is assigned blue and referred to as "elec" along the right side of the chart. The mission and propulsion sensor groups are represented similarly, assigned orange and green, respectively. The x-axis contains the type of model. The models output floating-point numbers, and the training labels are integers; thus, to reasonably ascertain the accuracy of these models, we round the models' outputs to the nearest integer to compute classification accuracy.

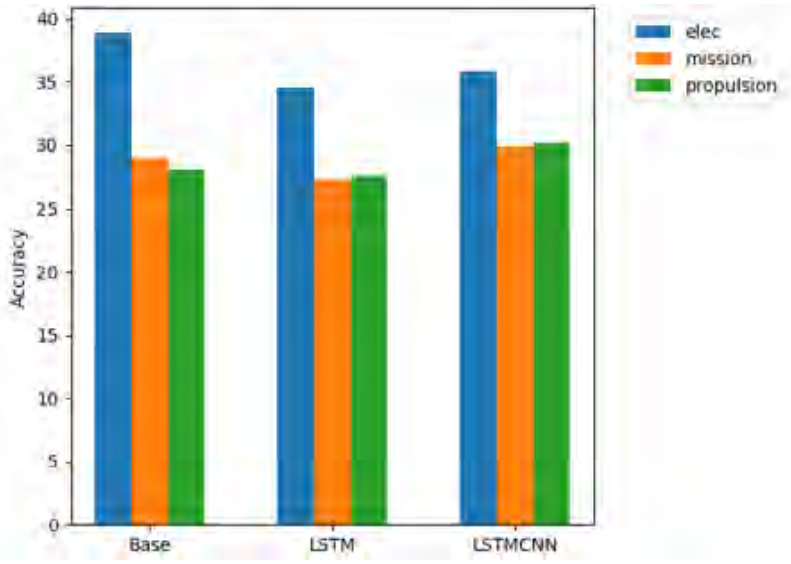


Fig.7: Average model accuracy for selected sensor groups, seq = 4, partition = A, minmax

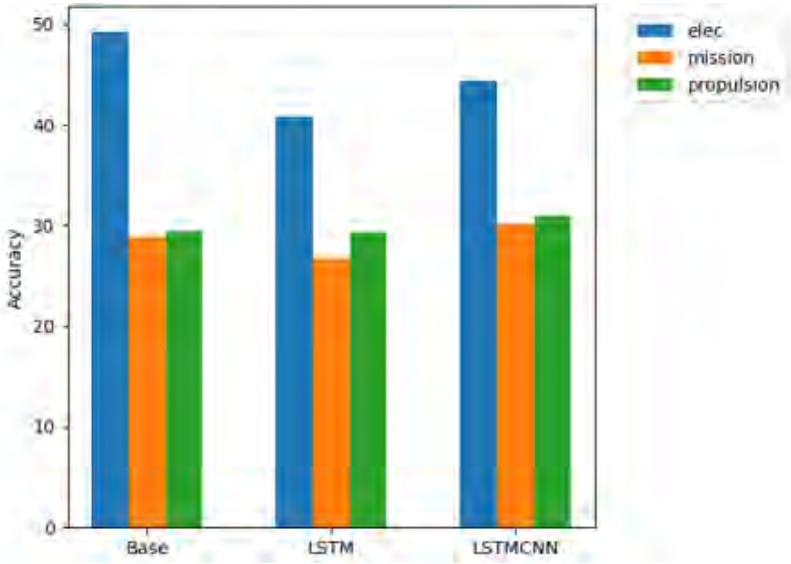


Fig.8: Average model accuracy for selected sensor groups, seq = 5, partition = A, minmax

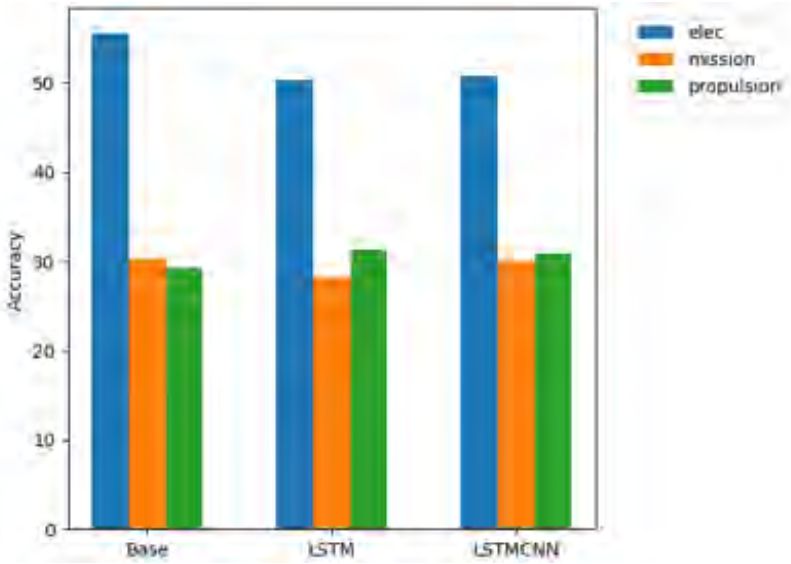


Fig.9: Average model accuracy for selected sensor groups, seq = 6, partition = A, minmax

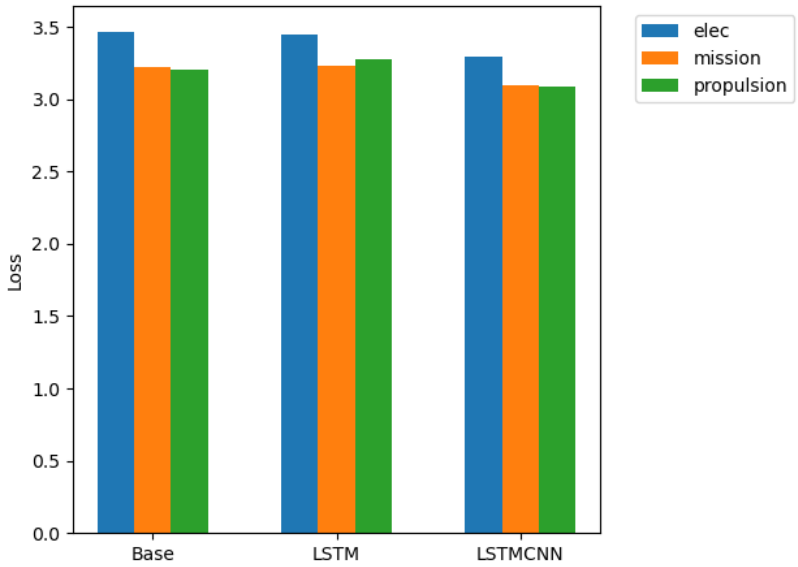


Fig.10: Average model loss for selected sensor groups, seq = 4, partition = A, minmax

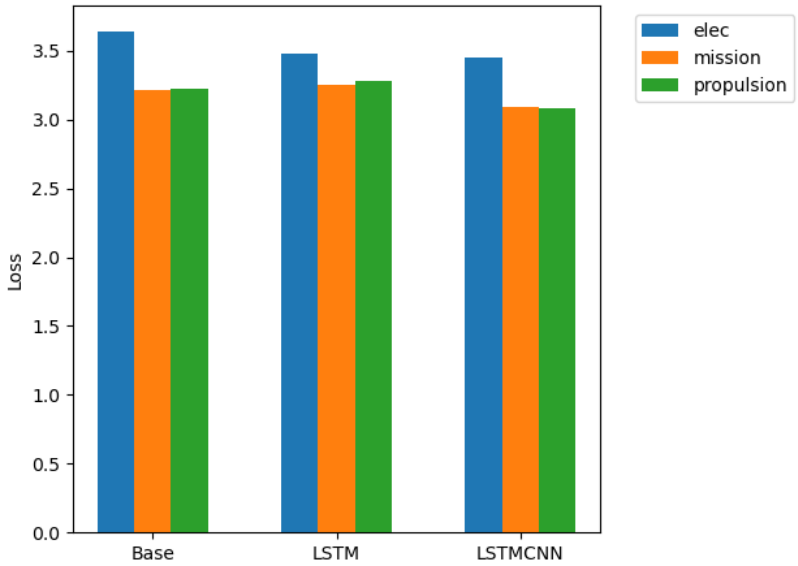


Fig.11: Average model loss for selected sensor groups, seq = 5, partition = A, minmax

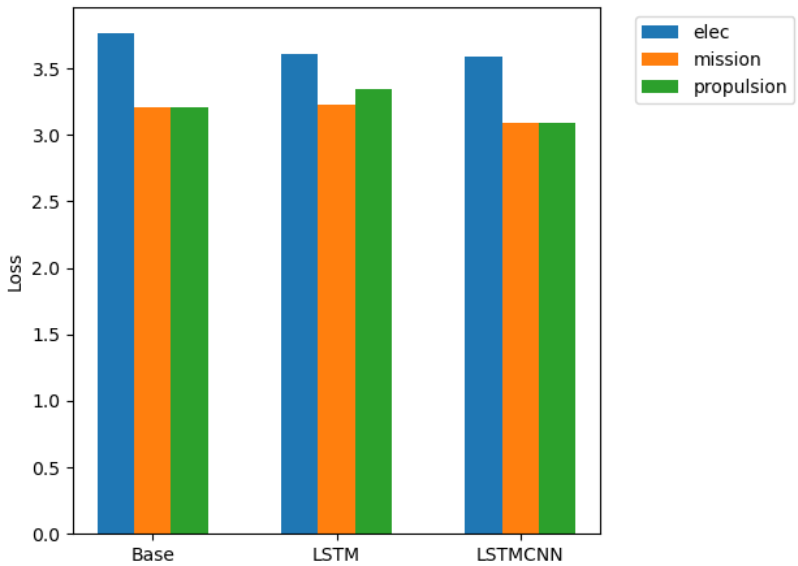


Fig.12: Average model loss for selected sensor groups, seq = 6, partition = A, minmax

Figs.7-9 demonstrate model accuracy when classifying the electrical, mission, and propulsion sensor groups, with each separate figure representing a different sequence length. Fig.10-12 demonstrate model loss with the same task.

Since it was determined that only the models trained on the electrical system were effective, new models were trained on datasets formed from combining sensor group 1 with the electrical system group. These are notated as “s1_g1” and “elec_s1_g1” respectively. The same process was applied to sensor groups 2 and 3. The newly trained models were then tested on both the new and old datasets. Their performances on each were then compared.

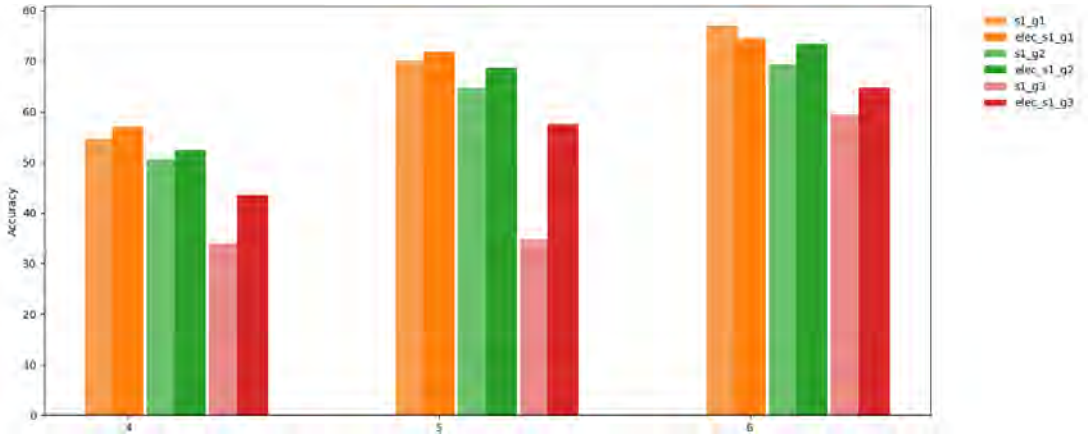


Fig.13: Average model accuracy between original and updated sensor groups - Base

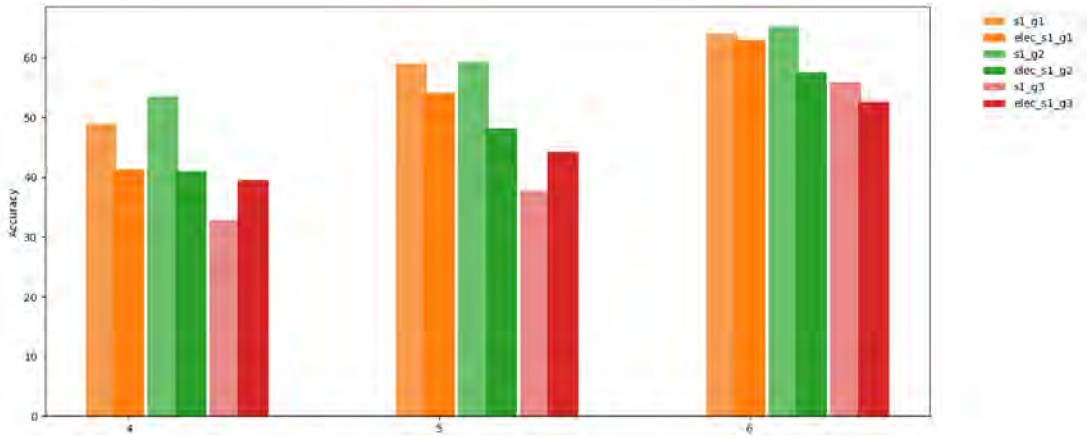


Fig.14: Average model accuracy between original and updated sensor groups - LSTM

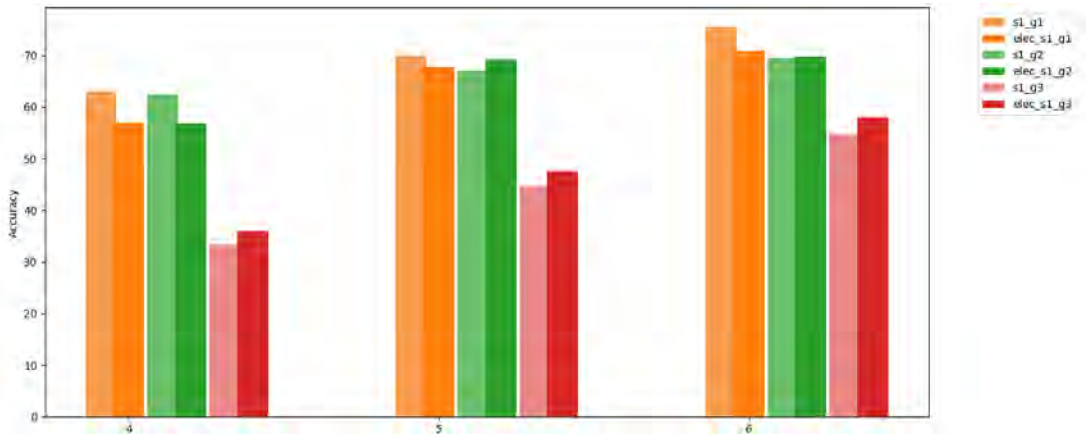


Fig.15: Average model accuracy between original and updated sensor groups - LSTMCNN

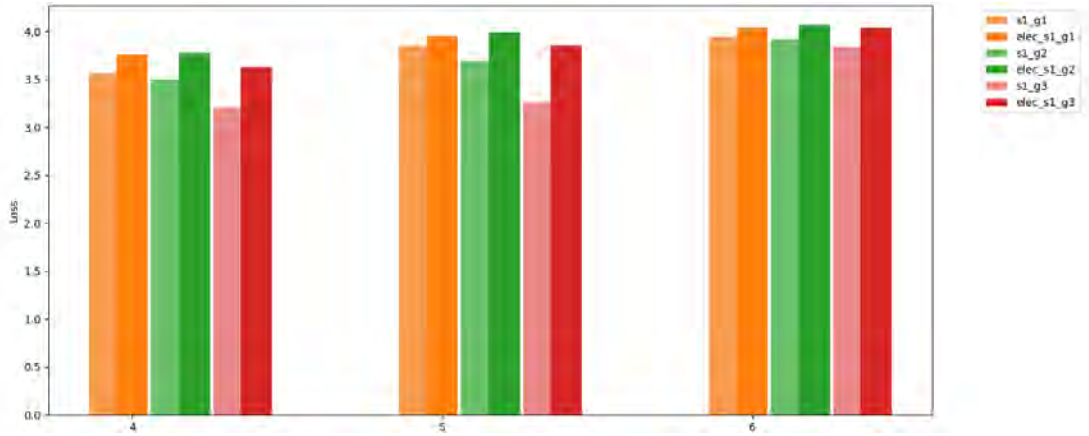


Fig.16: Average model loss between original and updated sensor groups - Base

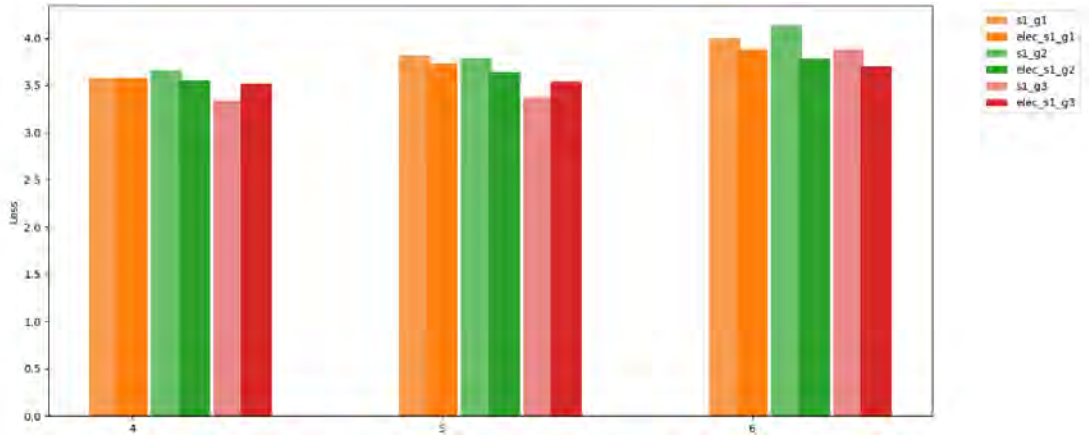


Fig.17: Average model loss between original and updated sensor groups – LSTM

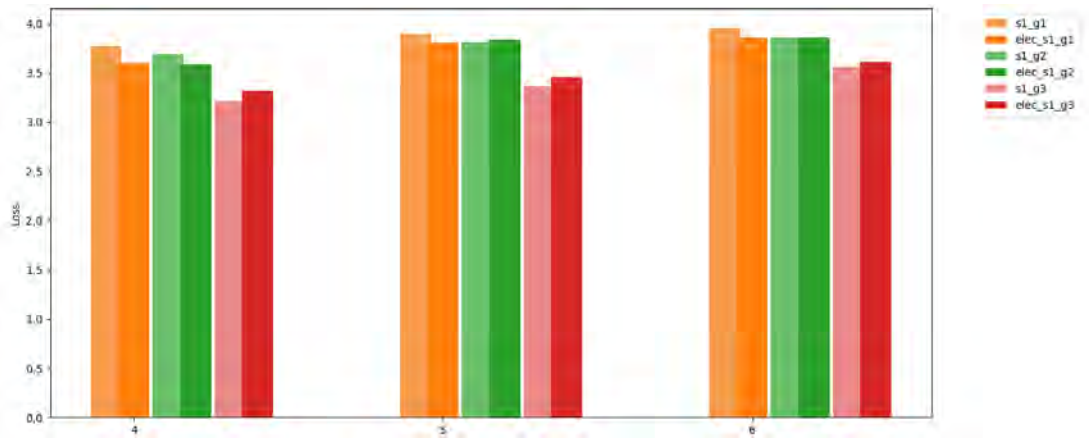


Fig.17: Average model loss between original and updated sensor groups - LSTMCNN

The charts in Figs.13-18 are presented with the following common features. Each chart showcases a singular model, Base, LSTM, or LSTMCNN, and compares its performance on two sets of sensor groups for varying sequence lengths. For each presented model, the data corresponding to input sensor group 1 is assigned orange and referred to as “s1_g1” along the right side of the chart, where “s1” refers to the first plant within the testbed and “g1” refers to group 1. A performance comparison on the new dataset is plotted next to its performance on the old dataset, indicated by a darker color. Sensor groups 2 and 3 are presented similarly to green and red, respectively. The x-axis details the original sequence length of the data. These models output floating-point numbers. To get a reasonable sense of the accuracy score for these models, we round their outputs to the nearest integer to compute the classification accuracy.

5. Discussion

The MEL's test bed continues to be an excellent environment for developing and testing RUL methodologies. Here, we examine the potential usefulness of the electrical, propulsion, and mission systems for RUL models.

5.1. Model Effectiveness on New Sensor Groups

As seen in Figs.7-9, the models trained on the electrical sensor group consistently outperformed the models trained on the mission and propulsion sensor groups in terms of average accuracy. The batch of mission and propulsion models was no better than guessing when observing their average accuracies. While the mission and propulsion models did not significantly change with the increase in sequence length, the electrical models improved with each increase in sequence length, performing the best on data with a sequence length of 6. Seeing that each model architecture improved with altering the sequence length for the electrical system but not for the mission or propulsion systems, it is indicated that the mission and propulsion systems may not be able to be observed accurately by the various model architectures, and thus these RUL techniques may not be effective for these systems.

When looking closer at the models' performances on the electrical system in Figs.7-12, we observe that while the average accuracy appeared to increase with an increase in the sequence length, the average loss stayed around 3.5 for each sequence length. A higher loss in this situation indicates that the incorrect classifications made by each model were incorrect by a somewhat large margin, and when the sequence length was increased, the incorrect classifications were off by an even larger margin. This is evident as the average accuracy was increasing, meaning that models were typically performing better, but the average loss was still hovering around 3.5, which could be explained by the models making more correct predictions on average but being incorrect by a larger margin on average as well.

5.2. Combined Sensor Groups

Once it was observed that the models only performed well on the electrical system, new models were trained on newly created datasets by combining old sensor groups with the electrical system dataset. After training this new batch of models, they were evaluated on the old sensor groups and the new, combined sensor groups. As seen in Figs.13-15, the average accuracies of the new models were similar when evaluated on both the old and new sensor groups. It should be noted that the models trained on the new group "elec_s1_g3" consistently performed better on the combined datasets than the old datasets, only being outperformed by the LSTM model on a sequence length of 6. We can infer from this that the features learned from the electrical datasets were not necessarily helpful when evaluating the s1_g3 dataset and combining these datasets may not be useful in the future when training new models. The models trained on this "elec_s1_g3" dataset also consistently underperformed on average accuracy compared to the other groups of models with their own respective datasets. This indicates that combining the electrical and s1_g3 sets and training models on the results does not necessarily lead to models that can learn the required representations to perform classification reliably.

Trends that were observed with previous models, which were only trained on singular sensor groups, were not observed in the newly trained models. Observing the previous trend associated with sequence length and average accuracy, we expect models with a sequence length of 4 to be the lowest performing and those with a sequence length of 6 to be the highest performing. However, the models trained on the combined datasets generally performed the worst on data with a sequence length 5. This is contrary to what is expected, as a longer sequence length means more information about the data, giving the model more information to learn from. This behavior may stem from the electrical data being transitional during sequence five and not as definitive as earlier or later data sequences, thus inhibiting model performance. This is worthy of examination in detail in a future case study.

6. Conclusion

This study has demonstrated the capabilities and impact of the Marine Engineering Lab's testbed for RUL prediction in maritime systems. It examined the efficacy of already-existing RUL evaluation models. It demonstrated their strengths and weaknesses, particularly in their abilities to accurately assess the electrical, mission, and propulsion systems in the MEL testbed for RUL method development.

Furthermore, the exploration of retraining models on combined datasets helped to examine the efficacy and future potential of utilizing more general models not limited to one system. The challenges and successes observed when training and testing models on datasets consisting of multiple sensor groups are valuable for the future of RUL prediction. Model accuracy for the combined sensors was encouraging, but the newly trained models introduced differences from expected trends that merit further examination in the future.

Overall, these insights provide a thorough look into the future of machine learning concepts and applications in the maritime field. Additionally, they offer a view into model generalization and complexity, as well as the potential challenges and successes that may result from further exploration.

References

BARU, A.; JOHNSON, R. (2023), *Three ways to estimate remaining useful life for predictive maintenance*, MathWorks. https://www.mathworks.com/company/technical-articles/three-ways-to-estimate-remaining-useful-life-for-predictive-maintenance.html

DAS, S.; HARRISON, G.A.; BODKIN, M.A.; HALL, R.; HERZOG, S. (2010), *Adaptive prognostic approaches combining regime identification with equipment operating history*, IEEE AUTOTEST-CON, pp.1-6

HOCHREITER, S.; SCHMIDHUBER, J. (1997), *Long Short-Term Memory*, Neural Computation 9, pp.1735-1780

KINGMA, D.P.; BA, J. (2015), *Adam: A Method for Stochastic Optimization*, 3rd Int. Conf. Learning Representations (ICLR)

KONGSBERG (2025a), *Predictive maintenance solution provides peace of mind for heavy lifting*, Kongsberg Maritime, https://www.kongsberg.com/maritime/news-and-events/our-stories/heavy-lifting/

KONGSBERG. (2025b), *Thruster remaining useful life: A predictive maintenance solution for our thrusters*, Kongsberg Maritime, https://www.kongsberg.com/maritime/services/kongsberg-remote-services/thruster-remaining-useful-life/

LECUN, Y.; BOSER, B.; DENKER, J.S.; HENDERSON, D.; HOWARD, R.E.; HUBBARD, W.; JACKEL, L.D. (1989), *Backpropagation Applied to Handwritten Zip Code Recognition*, Neural Computation 1(4), pp.541–551

LI, J.; LI, X.; HE, D. (2019), *A Directed Acyclic Graph Network Combined with CNN and LSTM for Remaining Useful Life Prediction*, IEEE Access 7, pp.75464-75475

LI, S.; ZHANG, C.; LIU, L.; ZHANG, X. (2024), *Gated Transient Fluctuation Dual Attention Unit Network for Long-Term Remaining Useful Life Prediction of Rotating Machinery Using IIoT*, IEEE Internet of Things J. 11, pp.18593-18604

MANOHAR, A.; DOWLING, M.; ARRIGAN, C.; DE MARTINIS, C.; SINGER, J. (2025), *Exploring Data-Driven RUL Methods for Marine Systems Using a Lab-Scale Ship Machinery Plant*, 17th HIPER Symp., Tullamore, http://data.hiper-conf.info/Hiper2025_Tullamore.pdf

MULAY, A.; MAJALI, A.; IYENGAR, V.; NAYAK, A.N.; SINGRU, P.M. (2022), *Fault identification and remaining useful life prediction of bearings using Poincare maps, fast Fourier transform and convolutional neural networks*, Mathematical Models in Engineering

NAIR, S.; VERMA, T.; KHATRI, R. (2019), *A Hybrid Model to Predict Remaining Useful Life for a Ball Bearing*, IEEE Symp. Series on Computational Intelligence (SSCI), pp.2119-2123

OLSON, S.A. (2024), *Creating a multi-model artificial intelligence framework to predict the operational availability of a laboratory-scale ship machinery plant*, Ph.D. Thesis, University of Michigan, Ann Arbor

OLSON, S.A.; McCOY, T. (2022), *Development of a multi-physics ship machinery systems laboratory for autonomous machinery research*, Advanced Machinery Technology Symp., Philadelphia

PASZKE, A.; GROSS, S.; MASSA, F.; LERER, A.; BRADBURY, J.; CHANAN, G.; KILLEEN, T.; LIN, Z.; GIMELSHEIN, N.; ANTIGA, L.; DESMAISON, A.; KÖPF, A.; YANG, E.; DEVITO, Z.; RAISON, M.; TEJANI, A.; CHILAMKURTHY, S.; STEINER, B.; FANG, L.; ET AL. (2019), *PyTorch: An Imperative Style, High-Performance Deep Learning Library*, ArXiv. https://arxiv.org/abs/1912.01703

REN, L.; DONG, J.; WANG, X.; MENG, L.; ZHAO, L.; DEEN, M. J. (2021), *A Data-Driven Auto-CNN-LSTM Prediction Model for Lithium-Ion Battery Remaining Useful Life*, IEEE Trans. Industrial Informatics 17(5), pp.3478-3487

VELASCO-GALLEGO, CHRISTIAN; LAZAKIS, IRALKIS. (2023), *Mar-RUL: A Remaining Useful Life Prediction Approach for Fault Prognostics of Marine Machinery*, Applied Ocean Research 140

YANG, JINSONG; LIU, YING; WEN, YANG; HUANG, XINYU. (2022), *Remaining Useful Life Prediction Method for Bearings Based on LSTM with Uncertainty Quantification*, Sensors 22(12)

ZHANG, G.; LIANG, W.; SHE, B.; TIAN, F. (2021), *Rotating Machinery Remaining Useful Life Prediction Scheme Using Deep-Learning-Based Health Indicator and a New RVM*, Shock and Vibration 2021(1)

Enhancing Marine Autonomous Vehicle Robustness Through AI-Driven Fault Discovery in Simulated Environments

Sean Hickey, University of Michigan, Ann Arbor/USA, hicsea@umich.edu
Joseph Serpa, University of Michigan, Ann Arbor/USA, jserpa@umich.edu
Jonathan Page, University of Michigan, Ann Arbor/USA, jonpage@umich.edu
Nickolas Vlahopoulos, University of Michigan, Ann Arbor/USA, nickvl@umich.edu

Abstract

The verification and validation of autonomous maritime systems represent a significant portion of their lifecycle cost, driven by system complexity and operation in dynamic environments. Conventional testing strategies often depend on costly physical trials that may not uncover rare yet critical system-level faults. The approach in this paper introduces an AI-driven methodology for automated fault discovery within a high-fidelity digital twin of a surface vessel. By employing a multi-objective search algorithm, the method co-evolves thousands of challenging operational scenarios and targeted fault injections, such as communication link disruptions. The search is guided by dual objectives for maximizing fault severity and test case diversity, ensuring a comprehensive exploration of the failure space. This technique enhances robotic system robustness by proactively identifying emergent faults, leading to improved operational safety and a significant reduction in validation costs.

1. Introduction

Ensuring the safety of autonomous maritime systems requires uncovering rare but hazardous interaction faults that conventional testing often misses. Achieving this efficiently calls for automated generation of diverse operational scenarios and realistic fault conditions. This paper introduces a flexible, multi-objective framework that applies a genetic algorithm (GA) to co-evolve scenario parameters and targeted fault injections. The framework directs the search toward severe failures while preserving test case diversity. It aims to be platform-agnostic by supporting systems that use a single launch point and modifiable configuration files. A plugin architecture allows users to extend the framework to new simulators, configuration formats, or messaging interfaces without extensive rework and customization. Users define custom termination conditions that specify success and failure criteria, enabling the framework to automatically classify outcomes across diverse simulation environments. The system also supports fault injection across multiple communication protocols, enabling extensive testing of message passing failures, *Natella et al. (2016)*, *Arlat et al. (2003)*, *Holzmann et al. (2023)*, *Jha et al. (2019*). Users can define domain-specific constraints to ensure the GA generates only physically realizable test cases, *Ding et al. (2023).*

The multi-objective search, using the NSGA-II algorithm, *Deb et al. (2002)*, simultaneously optimizes for fault severity and test input diversity, producing a diverse set of test cases while encouraging an even distribution among the possible termination conditions, *Feldt et al. (2008)*. The fault injection capability supports multiple fault types, including message dropping, delay, and modification, providing broad coverage of potential failure mechanisms. Real-time termination monitoring automatically classifies failure modes and guides the evolutionary search toward unexplored regions of the failure space.

The framework has been validated across multiple autonomous vehicle domains, from a custom in-house ground vehicle simulator, TankSim, *Hickey et al. (2024)*, implementing object avoidance and route following, to the high-fidelity HoloOcean marine robotics platform, *Potokar et. al (2022)*. This diversity demonstrates the framework's ability to adapt to varied simulation environments while maintaining consistent test generation capabilities. Unlike approaches restricted to simulator-bound scenario generation, this method exercises full autonomy stacks across domains by jointly exploring operational scenarios and injected failures.

2. Background

The verification and validation of autonomous vehicle systems constitute one of the most resource-intensive phases of their development lifecycle, *Yoo and Harman (2012)*. This challenge is amplified by the cyber-physical nature of these systems, which must operate reliably across diverse environmental conditions while handling complex sensor inputs, actuator outputs, and computational constraints. Traditional testing approaches ranging from unit tests to hardware-in-the-loop simulations struggle to identify system-level failures that emerge only under specific combinations of operational parameters and environmental conditions, *Schuette and Waeltermann (2005)*, *Mullins et al. (2018)*, *Song et al. (2023)*. Search-Based Software Testing (SBST) has emerged as a promising approach by framing test case generation as an optimization problem, *Humeniuk et al. (2022)*. Meta-heuristic algorithms systematically explore the input space to maximize fault detection while maintaining computational efficiency, *Humeniuk et al. (2023)*, *Khamprapai et al. (2021)*. Recent advances apply GAs and other search methods to discover safety violations in autonomous driving systems and to adaptively mine failure scenarios, *Zong et al. (2024)*, *Li et al. (2024)*, *Zhong et al. (2023)*. However, existing SBST frameworks for autonomous systems exhibit notable limitations. First, they are typically tightly coupled to specific simulation platforms, limiting their applicability across different autonomous vehicle domains, *Gambi et al. (2019)*, *Mullins et al. (2018)*. Second, they lack seamless integration with modern Continuous Integration/Continuous Deployment (CI/CD) pipelines, creating bottlenecks in rapid iterative development cycles due to the extensive time and resources it takes to run full-scale simulations. Third, many approaches focus primarily on specific aspects such as trajectory planning or perception failures, rather than holistic system-level testing that spans all subsystems, *Holzmann et al. (2023)*, *Zhong et al. (2023)*, *Khalastchi and Kalech (2018)*.

The software development paradigm for autonomous vehicles has shifted toward agile methodologies, with CI/CD pipelines enabling rapid iteration and deployment. In this context, manual test scenario creation becomes a critical bottleneck since engineers are limited by testing budgets and must create test sets that are as efficient as possible for hardware-in-the-loop test events, *Ding et al. (2023)*, *Song et al. (2023)*. Automated testing tools typically require significant manual configuration and lack the intelligence to discover edge-case failures that manifest only under specific environmental and fault injection conditions, *Khamprapai et al. (2021)*, *Zhong et al. (2023)*. Existing testing frameworks often cannot adapt to the variety of simulation environments used across different autonomous vehicle domains, *Bagci Das and Birant (2023)*.

3. Framework Methodology

3.1. System Architecture

The framework is organized around a setup phase and a GA execution loop shown in Fig.1.

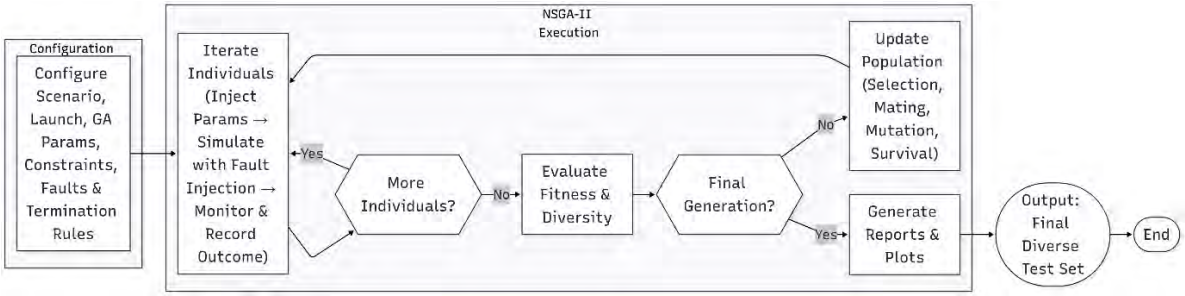


Fig.1: Condensed overview of framework workflow from setup through GA loop

In the setup phase, a user defines the launch configuration for starting the simulator including the launch script and configuration files, selects parameters from these configuration files that the GA will control, configures termination monitoring criteria, configures fault injection targets that become part of the GA search space, and establishes constraints for the simulator using the parameters from the configuration

files. Once these elements are defined, the GA operates in generations, where each population member is represented by a unique set of scenario parameters and fault injection parameters. For each candidate, the framework injects the configuration into the simulator's files, launches the run, applies termination monitoring, and collects results. These results feed into the GA's fitness evaluation, guiding the selection of the next generation.

3.2. Genetic Algorithm Configuration

The framework implements the Non-dominated Sorting Genetic Algorithm II (NSGA-II), a widely adopted approach for multi-objective optimization in autonomous systems, *Deb et al. (2002)*. Recent applications include safety violation detection in autonomous driving, *Zong et al. (2024)*, adaptive failure scenario mining, *Li et al. (2024)*, and underwater vehicle fault detection, *Bagci Das and Birant (2023)*.

The GA prioritizes two fitness functions: Error Severity Score (ESS) and maintaining input diversity so that failure modes are discovered in as many different ways as possible. In addition to maximizing the fitness values of each population member, the algorithm has been adapted to use a custom survival method to encourage an even output distribution of the termination codes. To manage scenario complexity, the problem uses mixed variables for fault and environment parameters.

The GA parameters are derived from values in the user-selected simulator configuration files. Each parameter corresponds to a selected variable, with user-defined bounds that constrain the search to realistic ranges. Additional parameters specify fault injection details, such as what messages to intercept, when to inject, and which fault type to apply, so the algorithm can co-evolve environment conditions and fault patterns that jointly expose failures.

3.3. Fault Injection Strategies

Fault injection enables systematic testing of how autonomous systems respond to hardware failures, communication disruptions, and sensor malfunctions that are difficult or dangerous to reproduce physically, *Arlat et al. (2003)*. Recent work has explored approaches including machine learning-guided fault injection, *Jha et al. (2019)*, and reinforcement learning for failure pattern identification, *Moradi et al. (2023)*. These approaches often rely on learned models or probability distributions for generating fault injections but require training data and retraining when systems change. Because this framework prioritizes immediate testing without prior knowledge and easy integration with rapid development cycles, it uses the GA to systematically discover combinations of fault injections that expose system-level failures.

The framework implements three fundamental fault types replicating real-world failure modes. Message dropping simulates communication failures from network congestion or hardware faults. Delay injection models network latency and processing delays that destabilize control loops. Content modification replicates sensor drift, calibration errors, or adversarial inputs by altering message fields while preserving structure. In TankSim, modified position messages uncovered false positives in collision detection, demonstrating the need for a more robust collision detection system. Future work could expand to include corruption and duplication faults.

The key insight is that the GA discovers optimal fault combinations, revealing interaction effects that single-fault testing misses. While some approaches test faults in isolation or use random combinations, *Natella et al. (2016)*, the GA learns which combinations effectively expose failures. For example, dropping position messages while delaying obstacle detection triggers collisions, whereas either fault alone is handled safely. The chromosome encodes fault type, timing, duration, and magnitude, allowing the GA to evolve sophisticated multi-fault patterns. The protocol-agnostic design enables fault injection across different communication systems through plugins. The ROS2 plugin intercepts messages between nodes using topic remapping, while the UDP plugin operates at the socket level. In the systems tested, these mechanisms introduced minimal additional latency, allowing the message interception to

be active during the whole simulation, even when a fault is not being injected. Supporting new protocols requires implementing an adapted interception interface and extending the core fault injection framework, which requires no rework to the core framework. The GA's evolved fault strategies reveal failure modes by systematically exploring the injection space. Over generations, the population converges on diverse fault patterns exposing different vulnerabilities, providing concrete test cases for debugging and regression testing.

3.4. Termination Condition Monitoring

The termination monitoring system observes simulation execution in real-time to detect and classify outcomes according to user-defined criteria. By terminating simulations promptly upon meeting success or failure conditions, the system reduces computational overhead while providing the GA with fitness feedback for each test case. Users configure termination conditions through four components. First, they select a monitor type, such as file watching or process monitoring. Second, they specify what to watch, like a log file path or vehicle state variable. Third, they define the triggering condition, such as a pattern match or value threshold. Finally, they assign a classification and severity score, with values ranging from 0.0 to 1.0 based on criticality. For example. system crashes might receive the maximum severity of 1.0 as critical failures, while timeouts receive 0.6 since they indicate inefficiency rather than fundamental flaws.

The framework's extensibility allows users to implement custom monitor classes for domain-specific requirements. While standard monitors cover common needs through file watching and process monitoring, some systems require specialized monitoring. For HoloOcean, the existing file monitors could not directly observe vehicle state for collision detection and positional awareness. A custom termination monitor plugin was created that directly accessed the simulator's vehicle state. This custom monitor integrated cleanly with the GA's fitness evaluation, allowing the framework to optimize for domain-specific success and failure criteria without modification to the core system. A user encodes their domain expertise through termination conditions, and the GA learns which input combinations trigger each termination type. The termination data currently drives fitness evaluation and survival selection, with future work exploring its use in improving crossover operators by biasing reproduction toward parameter combinations that produce diverse failure modes.

3.5. Constraint Management

The framework provides configurable constraint management to ensure all test cases are valid and to let users easily create and modify constraints. Users define constraints over parameters extracted from configuration files. Constraints are stored in the project database and evaluated during the search by the GA. Custom constraints handle mixed variables, including continuous, integer, discrete, and binary parameters. Test cases that violate a constraint do not survive to the next generation of the GA, preventing invalid test cases from reaching the final test set.

4. Multi-Domain Application Results

4.1. Case 1: Ground Vehicle Simulator

TankSim is a ROS2 framework created to emulate the control module of full-scale simulators and to exercise the algorithm in a controlled setting, *Hickey et al. (2024).* It integrates launch, termination monitoring, and message level fault injection to enable automated evaluation. An initial test campaign identified six distinct failure modes. The initial and final population distributions among the termination codes are shown in Fig.2 and Fig.3. After fixes, a second campaign ran overnight with a population of 100 for 25 generations for a total of 2,500 evaluations. The search varied scenario parameters and selected fault injections jointly, and user-defined termination conditions classified outcomes. After fixes, three notable failures remained. One valid test produced an unreachable target, indicating a planning limitation rather than an invalid scenario. Another case trapped the vehicle between an obstacle and a wall instead of navigating around it, exposing a controller flaw and suggesting the need

for improvements to path planning, perception, or waypoint following. A third case produced a false positive collision when a fault injection modified pose or obstacle messages. The system believed the vehicle was inside the obstacle even though it was not, highlighting the need for more robust collision detection. Within the 2,500 evaluations, the algorithm produced a broad distribution of failure modes and high parameter space diversity, yielding diverse and reproducible test cases that supported targeted debugging and design decisions.

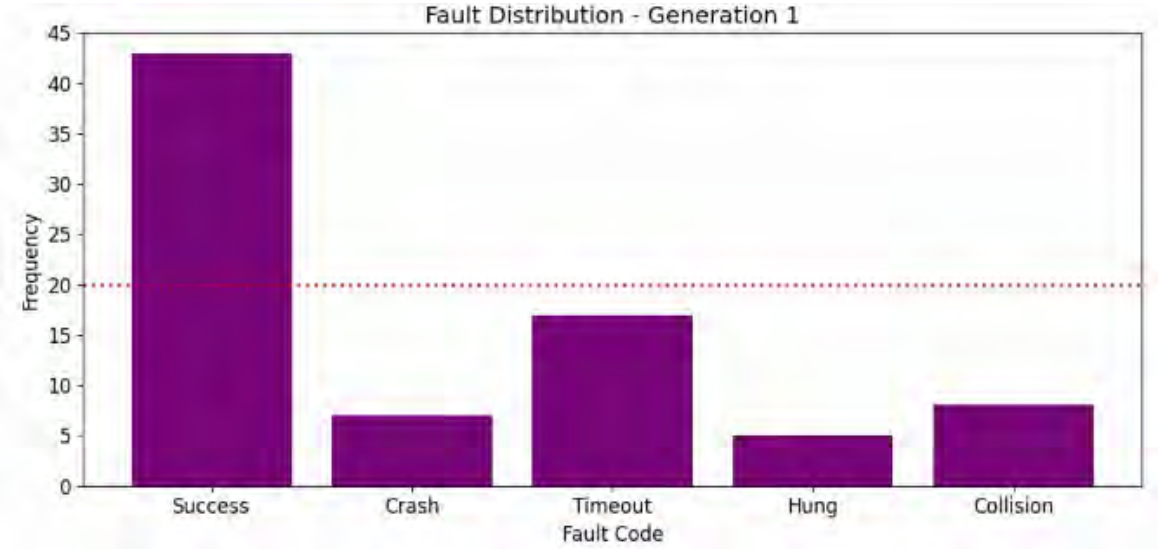


Fig.2: Initial termination distribution of initial test campaign, showing results of random search for comparison

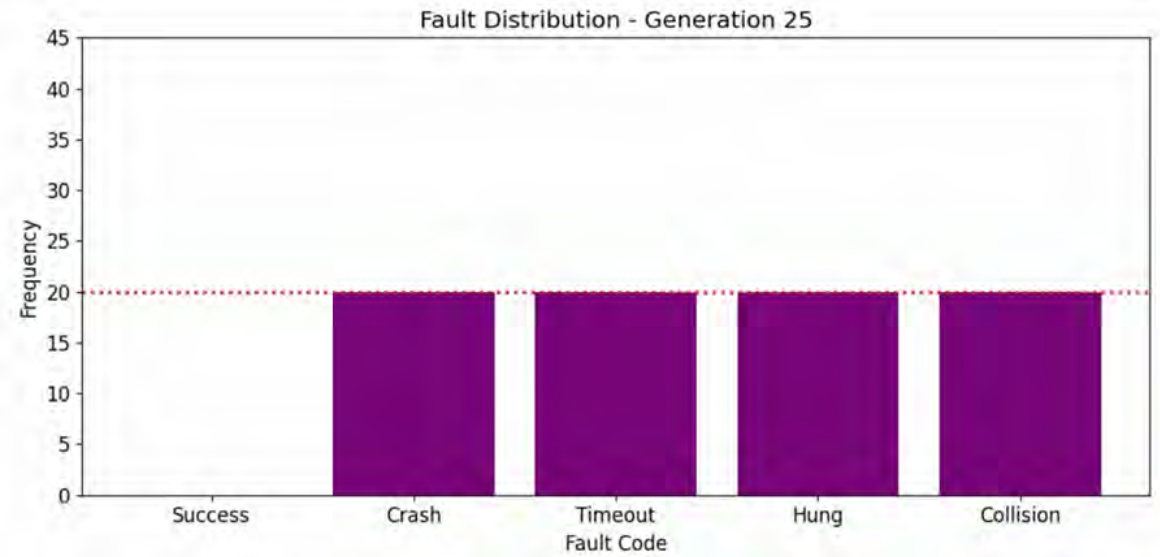


Fig.3: Final termination distribution of initial test campaign, showing results of GA search and displaying the desired even termination distribution

Fig.4 and Fig.5 demonstrate improved robustness over the initial test campaign, with the evidence of no failure modes found by random search, and some of the termination codes not being generated. Fig.6 demonstrates the effectiveness of the GA to maximize the fitness functions over the generations, maximizing the number of failure-inducing test cases in the test set, and maximizing the diversity of the parameter space. The input diversity eventually stops improving as rapidly over time due to the survival method selecting test cases to create the even output distribution.

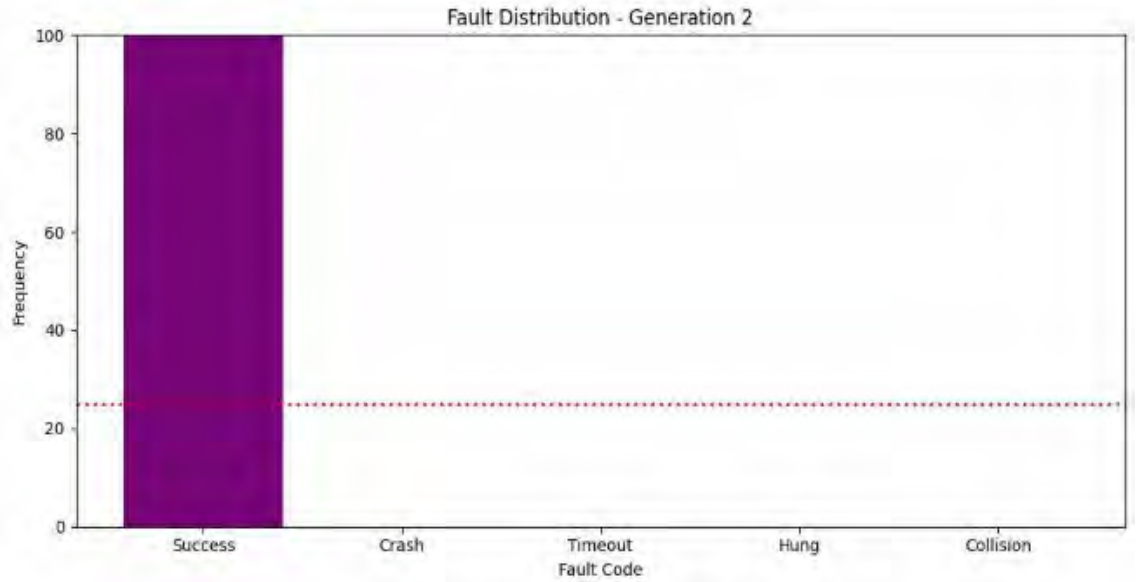


Fig.4: Initial termination distribution of second test campaign, showing results of random search for comparison (Shows Generation 2 because Generation 1 included constraint-violating cases)

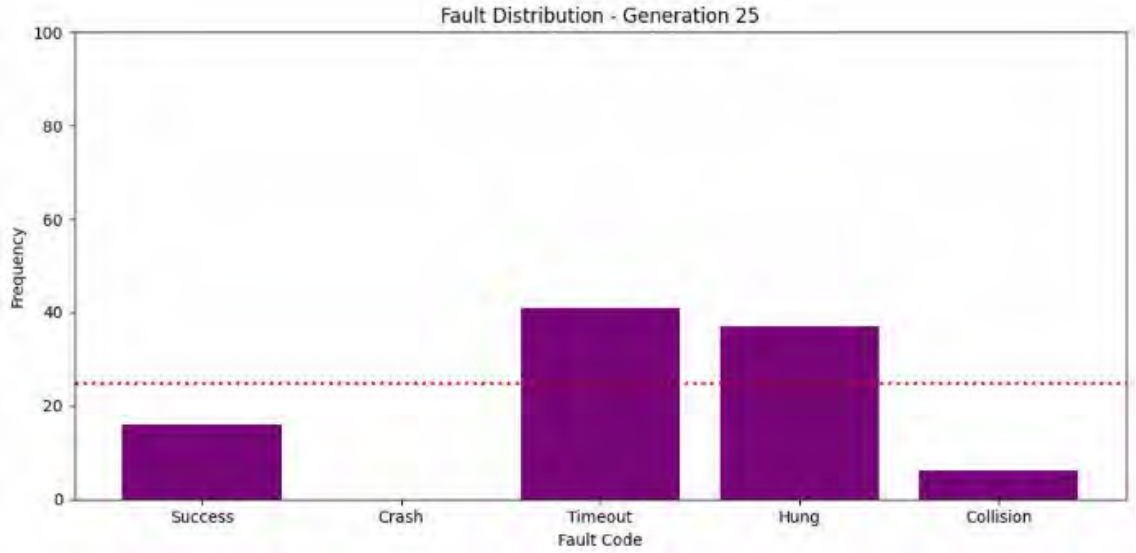


Fig.5: Final termination distribution of second test campaign, showing results of GA search and displaying a more robust system after initial failure modes were addressed

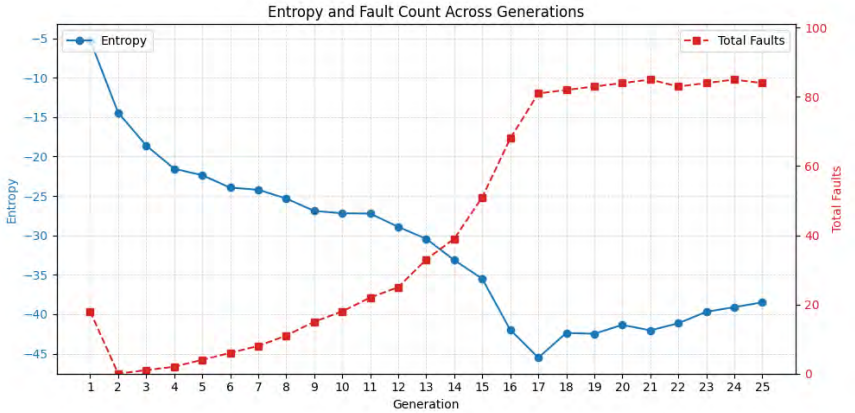


Fig.6: Fitness evaluation of the population over 25 generations in the second test campaign, demonstrating improved parameter space diversity and high rate of failure mode discovery

4.2. Case 2: Surface Vessel Simulator

HoloOcean is a high-fidelity marine simulator used here with a surface vessel in the PierHarbor world. The goal in using HoloOcean with this framework was to show how readily it can be integrated and automated. A small adapter handles launch and termination, and a custom state monitor classifies outcomes. GA parameters are defined by the user and stored in the project database, which keeps setup simple while preserving control over what the search explores. Concretely, we adapted HoloOcean by implementing a launch adapter that builds simulator commands from project configuration. We also added a termination monitor that queries the simulator's vehicle state to detect collision, grounding, or task completion events without modifying simulator internals.

For the initial experiment the parameters were encoded as the vessel's spawn pose (x, y, heading), target pose (x, y, heading), and a single static obstacle (x, y, radius) placed within PierHarbor, with bounds that enforce feasible spawns and obstacle placements. Because parameters are set in configuration files, this setup naturally extends to richer factors: environmental conditions (currents, wave state), waypoint sequences and mission rules, dynamic obstacles and traffic patterns, and sensor configurations such as sonar beam geometry, range, update rate, and noise models. Communication- and actuator-level parameters can also be exposed to study control robustness under perturbations. With a custom LCM injector, the framework could natively handle multi-agent scenarios. GA parameters can specify agent counts, roles, initial formations, and messaging schedules or drop and delay probabilities to explore faults between agents. HoloOcean supports custom vehicles and environments, so users can bring their own platforms and worlds. By mapping asset parameters into the configuration and mapping monitors into outcome definitions, these systems become testable in this framework through the same adapters.

Fig.7: Visual of a robotic surface vessel deployed automatically by the GA, (start position indicated by sphere) demonstrating an example of a test case generated by the algorithm.

5. Discussion and Future Work

The framework shows that automation and portability can be achieved with modest, localized effort. Using small adapters and project configuration, both TankSim and HoloOcean were brought into a single workflow. Termination monitoring was extended for HoloOcean, showing how the framework can run without altering core components. TankSim demonstrated that a fixed overnight budget can surface diverse and reproducible failures that led to fixes and regression tests, while the HoloOcean

integration showed that bringing a new simulator online is relatively straightforward once launch, parameter injection, and monitoring are connected through an adapter.

There are practical limits that shape how the system should be used. Results depend on simulator fidelity and on the quality of termination definitions, and compute budgets restrict how many evaluations fit into continuous testing windows. A genetic algorithm was chosen over alternatives such as Bayesian Networks or unsupervised learning for practicality in rapid, system-level testing. While Bayesian Networks can provide probabilistic insights into failure likelihood, they require substantial training data to learn parameter relationships before generating test cases. Unsupervised learning faces similar challenges, identifying patterns but not directly producing executable test cases. In contrast, genetic algorithms require no prior training, directly generate concrete test cases that engineers can execute, and adapt to code changes without retraining. With simulations taking seconds to minutes each, finding failures within 2,500 evaluations is practical for CI/CD integration. Alternative approaches would require comparable time for data collection plus model training.

Future work will deepen fault discovery and broaden coverage. The fault injection path will be extended to LCM for HoloOcean, and multiagent scenarios will be introduced to study communication failures. Domain-aware operators and adaptive mutation rates will be explored to improve search efficiency. Failure case minimization will be used to shrink counterexamples for faster analysis, and multi-fidelity workflows will combine fast surrogate runs with periodic high-fidelity checks. Finally, behavioral coverage metrics may be added as an explicit objective to steer exploration toward undertested behaviors.

6. Conclusion

Verification and validation of autonomous systems remain costly and time consuming, particularly when rapid iteration in CI/CD pipelines demands automated feedback at the system level. This work introduced a flexible, multi-objective search framework that treats test generation as optimization while remaining portable across simulators through small adapters and project configuration. The core separates launch, parameter injection, fault injection, and termination monitoring behind stable interfaces, and combines user defined termination with a constrained and diversity preserving NSGA-II search.

Application across two domains underscored these properties. In TankSim, an overnight run with a fixed budget uncovered diverse and reproducible failures, including unreachable targets, wedge behaviors around obstacles, and a false positive collision under an adversarial attack. These cases led to fixes and regression tests and thus improved system robustness. In HoloOcean, a surface vessel was brought online within the framework with a modest adapter and a custom state monitor, demonstrating that integration and automation require limited effort. Generating meaningful marine failures will involve configuring user defined GA parameters for this domain and extending the fault injection path to LCM.

The framework aligns with modern development practice. Runs are automated, budgets are explicit, artifacts are reproducible, and outputs can feed continuous testing and debugging. Future work will extend protocol coverage in HoloOcean, add multiagent scenarios to evaluate communication faults, explore adaptive mating and crossover to improve search efficiency, and apply the framework to other full-scale autonomous system simulators. Together, these steps aim to further close the gap between rapid iteration and robust system level assurance.

References

ARLAT, J.; CROUZET, Y.; KARLSSON, J.; FOLKESSON, P.; FUCHS, E.; LEBER, G.H. (2003), *Comparison of physical and software-implemented fault injection techniques*, IEEE Trans. Computers 52/9, pp.1115-1133

BAGCI DAS, D.; BIRANT, D. (2023), *GASEL: Genetic algorithm-supported ensemble learning for fault detection in autonomous underwater vehicles*, Ocean Eng. 272

DEB, K.; PRATAP, A.; AGARWAL, S.; MEYARIVAN, T. (2002), *A fast and elitist multiobjective genetic algorithm: NSGA-II*, IEEE Trans. Evolutionary Computation 6/2, pp.182-197

DING, W.; XU, C.; ARIEF, M.; LIN, H.; LI, B.; ZHAO, D. (2023), *A Survey on Safety-Critical Driving Scenario Generation -- A Methodological Perspective*, arXiv:2202.02215

FELDT, R.; TORKAR, R.; GORSCHEK, T.; AFZAL, W. (2008), *Searching for cognitively diverse tests: Towards universal test diversity metrics*, IEEE Int. Conf. Software Testing, Verification and Validation Workshops, Lillehammer

GAMBI, A.; MUELLER, M.; FRASER, G. (2019), *Automatically testing self-driving cars with search-based procedural content generation*, ACM SIGSOFT Int. Symp. Software Testing and Analysis (ISSTA), Beijing

HICKEY, S.; VLAHOPOULOS, N.; SMEREKA, J.; ZHANG, G. (2024), *Elements of unsupervised testing for software systems of autonomous vehicles*, Interservice/Industry Training, Simulation, and Education Conf. (I/ITSEC), Paper No. 24346

HOLZMANN, H.; LANDERSHEIM, V.; PIRAM, U.; BARTOLOZZI, R.; STOLL, G.; ATZRODT, H. (2023), *Fault Injection in Actuator Models for Testing of Automated Driving Functions*, Vehicles 5/1, pp.94-110

HUMENIUK, D.; KHOMH, F.; ANTONIOL, G. (2022), *A search-based framework for automatic generation of testing environments for cyber-physical systems*, Information and Software Technology 149

HUMENIUK, D.; KHOMH, F.; ANTONIOL, G. (2023), *AmbieGen: A search-based framework for autonomous systems testing*, Science of Computer Programming 230

JHA, S.; BANERJEE, S.S.; TSAI, T.; HARI, S.K.S.; SULLIVAN, M.B.; KALBARCZYK, Z.T.; KECKLER, S.W.; IYER, R.K. (2019), *ML-based Fault Injection for Autonomous Vehicles: A Case for Bayesian Fault Injection*, arXiv:1907.01051.

KHALASTCHI, E.; KALECH, M. (2018), *On Fault Detection and Diagnosis in Robotic Systems*, ACM Computing Surveys 51/1, Article 9

KHAMPRAPAI, W.; TSAI, C.-F.; WANG, P.; TSAI, C.-E. (2021), *Performance of Enhanced Multiple-Searching Genetic Algorithm for Test Case Generation in Software Testing*, Mathematics 9/15, 1779

KLIKOVITS, S.; CASTELLANO, E.; CETINKAYA, A.; ARCAINI, P. (2023), *Frenetic-lib: An extensible framework for search-based generation of road structures for ADS testing*, Science of Computer Programming 230

LI, Y.; WU, S.; WANG, H. (2024), *Adaptive Mining of Failure Scenarios for Autonomous Driving Systems Based on Multi-population Genetic Algorithm*, IEEE Intelligent Vehicles Symp. (IV), Seoul

MENGHI, C.; NEJATI, S.; BRIAND, L.; PARACHE, Y.I. (2020), *Approximation-refinement testing of compute-intensive cyber-physical models: an approach based on system identification*, ACM/IEEE Int. Conf. Software Engineering (ICSE), Seoul

MORADI, M.; VAN ACKER, B.; DENIL, J. (2023), *Failure Identification Using Model-Implemented Fault Injection with Domain Knowledge-Guided Reinforcement Learning*, Sensors 23/4

MULLINS, G.E.; STANKIEWICZ, P.G.; HAWTHORNE, R.C.; GUPTA, S.K. (2018), *Adaptive generation of challenging scenarios for testing and evaluation of autonomous vehicles*, J. Systems and Software 137, pp.197-215

NATELLA, R.; COTRONEO, D.; MADEIRA, H.S. (2016), *Assessing Dependability with Software Fault Injection: A Survey*, ACM Computing Surveys 48/3, Article 44

POTOKAR, E.; ASHFORD, S.; KAESS, M.; MANGELSON, J.G. (2022), *HoloOcean: An Underwater Robotics Simulator*, IEEE Int. Conf. Robotics and Automation (ICRA), Philadelphia, pp.3040-3046,

SCHUETTE, H.; WAELTERMANN, P. (2005), *Hardware-in-the-Loop Testing of Vehicle Dynamics Controllers - A Technical Survey*, SAE Trans. 114, pp.593-609

SONG, Q.; TAN, K.; RUNESON, P.; PERSSON, S. (2023), *Critical scenario identification for realistic testing of autonomous driving systems*, Software Quality J. 31/2, pp.441-469

YOO, S.; HARMAN, M. (2012), *Regression testing minimization, selection and prioritization: A survey*, Software Testing, Verification and Reliability 22/2, pp.67-120

ZHONG, Z.; KAISER, G.; RAY, B. (2023), *Neural Network Guided Evolutionary Fuzzing for Finding Traffic Violations of Autonomous Vehicles*, IEEE Trans. Software Eng. 49/4, pp.1860-1875

ZONG, H.; HOU, Z.; LIU, H. (2024), *Safety-Violation Scenarios Search for ADS via Multi-Objective Genetic Algorithm*, IEEE Conf.

Self-Propulsion CFD Simulation of a Bulk Carrier Vessel with and without Pre-Swirl Stators (PSS)

Alex Shiri, RISE, Gothenburg/Sweden, alex.shiri@ri.se
Dmitry Ponkratov, SIEMENS, London/UK, dmitry.ponkratov@siemens.com
Qingshan Zhang, COSCO Shipping, Shanghai/China, zhang.qingshan@coscoshipping.com
Wie Jin, Everllence, Copenhagen/Denmark, jin.wei@everllence.com
Dong Cheol Seo, NRC-CNRC, Ottawa/Canada, Dong.Seo@nrc-cnrc.gc.ca
Yan Xing-Kaeding, HSVA, Hamburg/Germany, xing-kaeding@hsva.de
Themistoklis Melissaris, Wärtsilä, Eindhoven/Netherlands, themis.melissaris@wartsila.com
Rikard Johansson, Kongsberg, Kristinehamn/Sweden, rikard.johansson@km.kongsberg.com
Ram Kumar Joga, Indian Register of Shipping, Mumbai/India, J.Kumar@irclass.org

Abstract

This study presents the results of a workshop conducted within the JoRes framework, focusing on self-propulsion CFD simulation of the full-scale bulk carrier vessel, JoRes5 GRIP. This vessel serves as an ideal validation case, having undergone sea trials both with and without an Energy-Saving Device (ESD)—specifically, pre-swirl stators (PSS). The sea trials were performed at comparable draught conditions (TM = 7.7 m) and across multiple speeds, providing a robust comparative dataset. The vessel, with a length between perpendiculars (LPP) of 182 m, was tested at speeds of 13.65, 15.25, 16.02, and 16.32 kn. These conditions were replicated in the CFD simulations conducted during the workshop, providing insights into the influence of PSS on propulsion performance and enabling validation against full-scale measurement.

1. Introduction

Continuous advancements in computational fluid dynamics (CFD) have equipped researchers with increasingly accurate tools for predicting the performance of full-scale vessels. However, to ensure reliability, these computational predictions must be validated against high-quality sea trial data. Within the JoRes Joint Research Project, https://jores.net/, a comprehensive dataset of sea trials has been developed to benchmark various vessel types and support efforts to improve energy efficiency in maritime transport.

This study presents the outcome of a workshop conducted under the JoRes framework, focusing on the self-propulsion CFD simulation of the full-scale bulk carrier JoRes5 GRIP. This vessel provides an ideal validation case, having undergone sea trials both with and without an energy-saving device (ESD)—specifically, pre-swirl stators (PSS). The trials were conducted at a consistent draught of TM = 7.7 m and at multiple speeds, offering a robust dataset for comparative analysis. The vessel was tested at speeds of 13.65, 15.25, 16.02, and 16.32 kn, and these conditions were replicated in the CFD simulations performed during the workshop.

The primary objective of the workshop was to numerically simulate the sea trials using CFD for validation purposes. Previous comparative analysis of different CFD codes and numerical approaches used for self-propulsion simulations revealed improvements in the accuracy of power prediction. Nonetheless, further refinement is needed to enhance consistency and reliability. This paper also outlines the evaluation of simulation results, with a particular focus on estimating the power gain associated with the use of ESDs such as PSS.

2. Why is this case unique?

Energy-saving devices have been introduced in the maritime industry for some time, but there are still ongoing debates about whether their installation is worthwhile. The challenge is associated with an explicit confirmation of fuel saving due to the device installation. If the device is installed on a newly

built vessel, there is no sea trial record without the device. Hence, it is impossible to understand the contribution of savings made possible by the device. It appears that this challenge can be addressed on a retrofitted vessel, as trials could be performed before entering the dry dock (without an energy-saving device) and then after the dry dock (with an energy-saving device).

In practice, to save costs, the dry-docking opportunity is also used not only for the ESD installation but also for hull cleaning and new paint application. As a result, when the vessel is out of the dock, there are at least two significant alterations (a clean hull with new paint and the ESD) that make it difficult to understand the specific contribution to the savings from ESD. Another opportunity is to compare long-term monitoring data for the vessel with and without ESD; however, if the operational conditions (draughts, speeds, duration of voyages, and idle times) are not the same, it introduces uncertainty into the comparison.

For the JoRes5 GRIP bulker case, the story is different and somewhat unique. The vessel was launched late September (the vessel's identity is anonymous, so the year of launch is not disclosed) and remained alongside the harbour until the beginning of April. Because she was in the water in the winter months, it was expected that not too much marine growth would appear on the hull. By the end of March, the propeller was polished by divers, and the first trials without the ESD were performed. After the first trial, the ship entered dry dock for the installation of the PSS. Upon arrival in dry dock two weeks later, the vessel was inspected immediately, before any cleaning or painting took place. As expected, the hull was found in an immaculate condition, with no visual traces of green weed fouling. Only a thin layer of slime was present in certain areas. The effect of the slime layer is considered minimal; the ship had a very low draught with approximately 5 m aft and 0.1 m fore during the six-month winter period at the yard. Only the bottom surface was exposed to water, where no fouling had developed due to the absence of sunlight.

In the dock, the flat bottom was recoated with a second layer of anti-fouling coating. In theory, this changes the surface roughness of the bottom. However, as the existing paint layer was new and untouched, the effect on frictional resistance, and hence vessel performance, can be considered negligible. This exercise suggests that, unlike many previous attempts, the hull conditions for the trials with and without ESD were almost identical. Roughness measurements were not performed on this vessel; however, comparing the photos of the JoRes5 bulker and JoRes1 tanker in the docks show that they had a similar type of paint and similar paint conditions, so an assumption was suggested to use the tanker's roughness values for the bulker, too. It would also be helpful to develop good comparisons between the JoRes1 tanker and the JoRes5 bulker, which also have similar dimensions.

3. Vessel Description and Sea-Trial Condition

The JoRes 5 GRIP is a 52,000 DWT bulk carrier with a length of 189.9 m and a beam of 32.26 m. The vessel is propelled by a diesel direct configuration using an engine that generates a power of 8,600 kW at 121 RPM. A picture of the vessel is provided in Fig.1. Fig.2 shows the installed Pre-Swirl-Stator (PSS) energy saving device (ESD).

The shaft power was measured using a torsion meter on the propeller shaft, which was based on strain gauges. For both trials, the same instrumentation and strain gauge on the propeller shaft were used to avoid bias errors. Position, course and speed over ground were determined using a DGPS unit, which was installed on the bridge top. The relative wind speed and direction were obtained using a sonic anemometer positioned on the mast on top of the wheelhouse. Data was stored automatically with a sampling frequency of 10 Hz on a PC located on the bridge. Furthermore, a visual recording was made of the draught, observing the draught markings on the hull from a small boat. The depth below the keel was manually recorded from the echo sounder indicator in the wheelhouse. The water depth at the trial location was 110 m. The displacement has been calculated using the visually recorded draught.

Fig.1: JoRes5 GRIP bulker during the sea trials

Fig.2: Pre-Swirl Stator (PSS) installed in the dry dock

To record the wave height, period and direction, a free-floating Datawell Directional Wave Buoy of type DWR G4 was used during trial 1 (without PSS). For trial 2 (with PSS installed), the wave height was estimated visually, as the wave buoy was not deployed due to miscommunication. However, the sea conditions were fair during the trials. Table ITable I presents an overview of the speed trials condition.

Double-speed runs of 10 minutes were conducted at four power settings. The heading of the runs was fixed at 151° and 331° to obtain sufficient space for turning and accelerating the vessel. This deviates from the ISO 15016 standard, which states that the heading of the vessel should be chosen in accordance with and against the dominant wave direction. However, as the waves were low (0.15 m), this had no significant effect on the wave corrections. The uncorrected performance data has been analysed and corrected for the non-ideal weather and environmental conditions according to the ISO 15016 standard.

The analysis has been performed using the STAIMO program version 1.2.0.

Table I: Overview of speed trials condition

Parameter	Trial 1	Trial 2	Unit
	Without PSS	With PSS	-
Length over all	189.9	189.9	m
Length between perpendiculars	182.0	182.0	m
Breadth	32.26	32.26	m
Moulded draught at forward perpendicular	7.70	7.68	m
Moulded draught at midship	7.75	7.73	m
Moulded draught at aft perpendicular	7.73	7.71	m
Moulded displacement	36142	36040	m^3
Midship area coefficient	0.994	0.994	-
Main section area	249	248	m^2
Wetted surface of hull	7173	7165	m^2
Wetted surface of rudder	99	99	m^2
Wetted surface of bilge keels	70	70	m^2
Bow length for wave correction	42	42	m
Height of anemometer	38	38	m
Transverse wind area	772	773	m^2
Average significant wave height	0.12	0.15	m
Average true wind speed	0.90	1.6	m/s

It was intended to perform both sets of trials (with and without PSS) under the same loading conditions. The difference in draught between the trials, therefore, was only 20 mm, Table I. Nevertheless, the middle draught is larger than the forward and aft draughts for both trials, and the vessel was sagging, Fig.3.

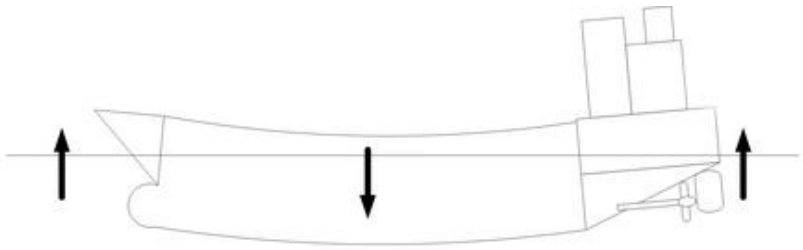


Fig.3: Vessel is sagging

It is normally assumed in CFD that the vessel is rigid and has no sagging or hogging. Uncertainty analysis was performed for this case to understand the effect of different components (including the change in draught) on the vessel performance. The uncertainties can be expressed by a 'Performance Indicator' (PI). The performance indicator 'PI' represents the relative power deviation at a constant speed from the reference speed/power condition. The reference speed/power condition is here taken as the hypothetical 'true' condition without measurement errors. The advantage of this method is that a single indicator can express uncertainties in both speed and power.

As shown in Fig.4, if the draught is estimated with insufficient accuracy (e.g., a 100 mm error at all marks), this will result in a performance difference of approximately 1%. In our case, the deviation between the draughts is only 20 mm, so it is expected that it will have a minor effect on the performance. Therefore, we suggest to use the forward and aft draught values and ignore the midship draught. In this

case, there will be a minor deviation in displacement; however, we will ensure a bit more realistic flow behaviour around the bow and stern.

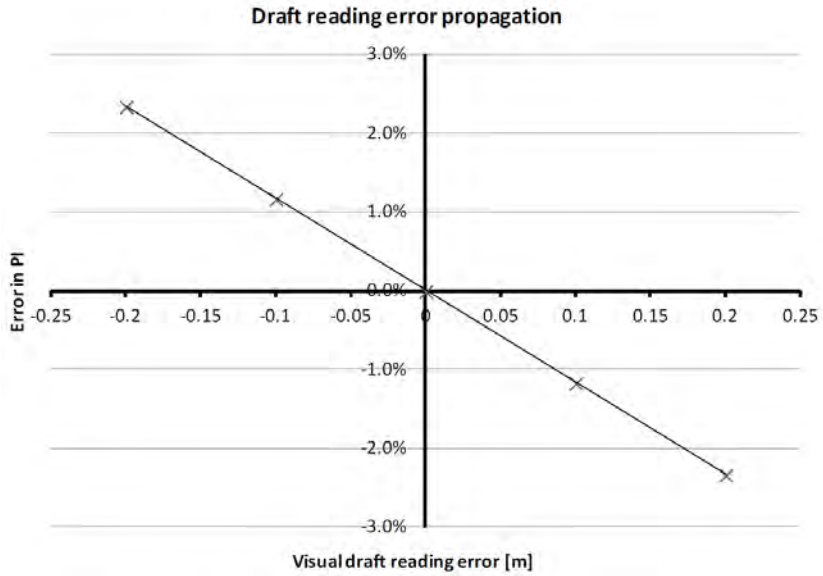


Fig.4: Performance Indicator (PI) error (%) vs draught reading error (m)

3. Sea Trial Result

The results show that to achieve a speed of 16 kn, a shaft power of 7975 kW is required with the PSS in place, which is 6.8% less power compared to the 8558 kW that is required to achieve the same speed without the PSS in place. Taking the uncertainty into account, it can be said that the difference between the trials is 6.8% ± 1%. At equal power, this corresponds to a speed difference of 0.3 knots, which is favourable for the condition with the pre-swirl stator in place.

The required RPM to achieve a shaft power of 7975 kW decreases from 124.0 to 120.3 RPM due to the presence of the PSS, Fig.5. The light running margin without PSS in place is 5.2%. After installation of the PSS, the light running margin has decreased to 2.0%.

The main objective of the ship scale verification was to perform the sea trials without and with PSS at the identical settings. Hence, the target was to keep the shaft RPM as close as possible to both sets of trials. Nevertheless, due to the PSS effect, the vessel speeds differed. Table II shows the speeds for no waves, no current, and no wind conditions after corrections according to ISO 15016.

Table II: Vessel speeds with and without PSS, while operating in similar shaft RPM

	Trials without PSS	Trials with PSS
Vessel speed 1	13.06 kn	13.65 kn
Vessel speed 2	14.61 kn	15.25 kn
Vessel speed 3	15.40 kn	16.02 kn
Vessel speed 4	16.01 kn	16.32 kn

Fig.5 shows the result of measured and corrected shaft power values at different vessel speeds. Fig.6 shows the corresponding shaft rotational speeds plotted against the measured shaft power. In both figures, the measured data have been curve-fitted to establish a reference for comparing power and RPM at given vessel speeds. In CFD self-propulsion simulations, the vessel speed is held constant while the propeller rotational speed (RPM) is adjusted to balance the resistance and thrust forces at the given speed. It may be inconvenient to run the cases with and without PSS at different speeds; so it was recommended to choose the speed in one set of trials and interpolate all the results for the other set to these speeds. Table III presents the corrected power and RPM for the same four speeds to be used in

CFD computations.

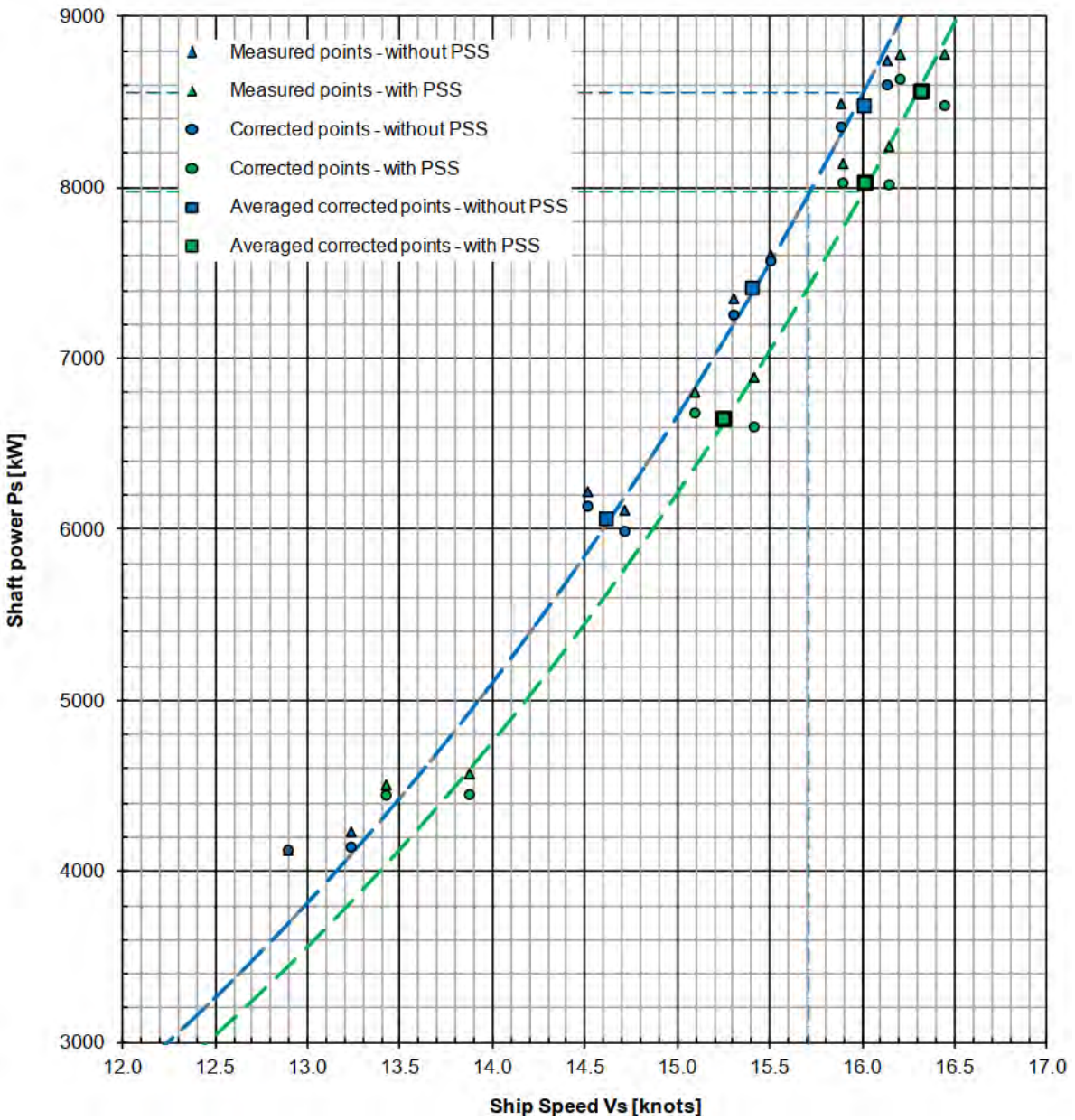


Fig.5: Measured and averaged corrected shaft power for different speeds of vessel at sea trial with and without PSS

Table III: Corrected and curve-fitted results of sea trial for 4 speeds to be used in CFD workshop

Speed [kn]	Power [kW]			RPM		
	Without PSS	With PSS	Diff.	Without PSS	With PSS	Diff.
13.65	4745	4408	-7.1%	105.1	99.8	-5.0%
15.25	7048	6579	-6.7%	119.5	113.4	-5.1%
16.02	8435	7947	-5.8%	126.6	120.5	-4.8%
16.32	9025	8476	-6.1%	129.4	123	-4.9%

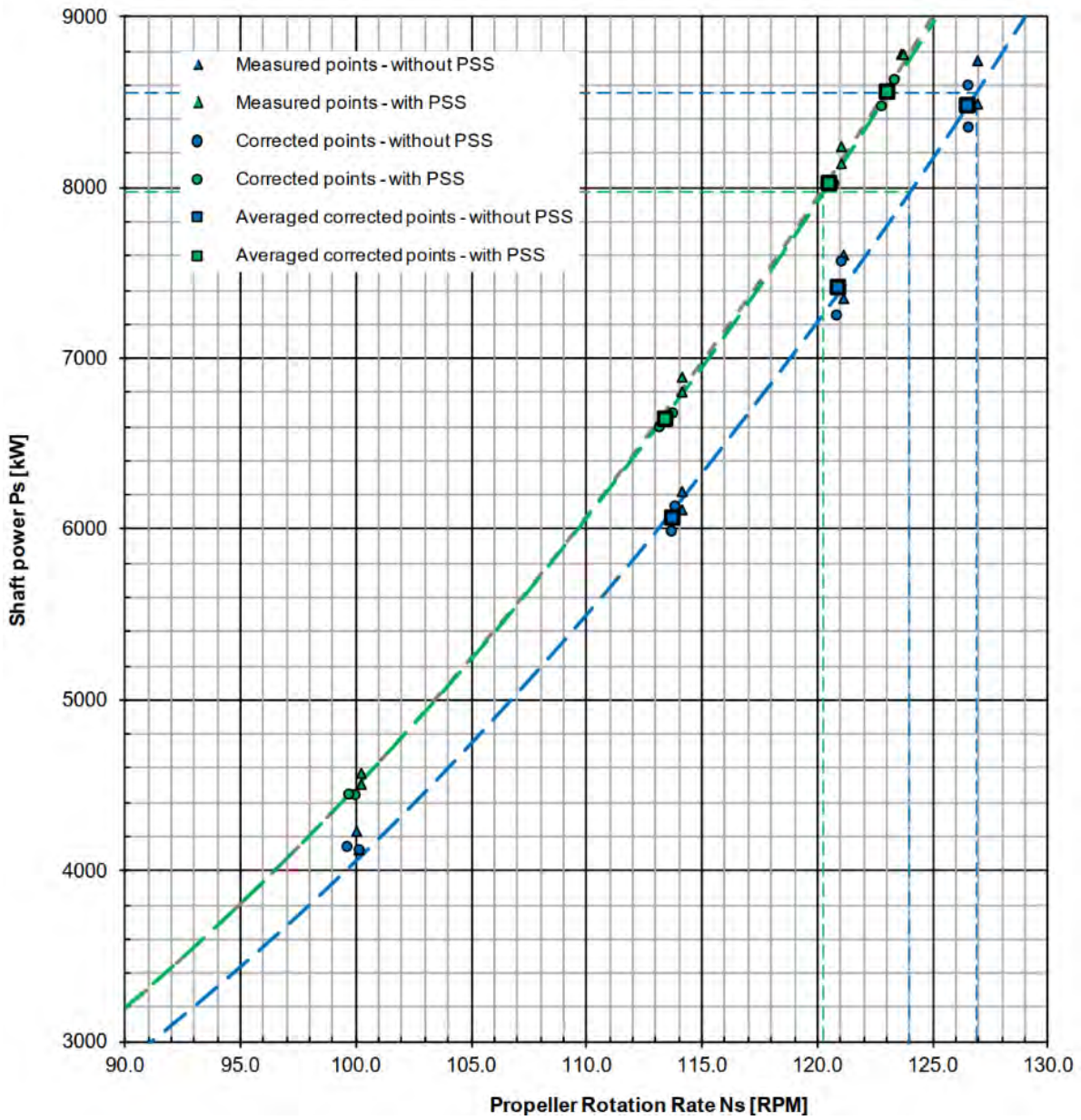


Fig.6: Shaft power vs. propeller rotation speed for vessel at sea trial with and without PSS

4. Self-Propulsion CFD simulation

Based on the sea trial report, a detailed case description was developed, specifying the necessary conditions such as vessel speeds, draughts, and hull surface roughness. To ensure an unbiased comparison, the RPM and shaft power values were not disclosed to participants, enabling a blind CFD validation against the measured data. The geometry files used in the simulations are publicly available on the JoRes website. The provided hull model includes the rudder, propeller hub, propeller blades, and configurations both with and without the pre-swirl stator (PSS). The principal particulars of the hull are listed in Table IV, and the propeller specifications are detailed in Table V.

The superstructure is not considered in the CFD simulation. Based on *ITTC (2008)* procedures, the total ship resistance coefficient could be written as:

$$C_T = \frac{S + S_{BK}}{S}\left[(1+k)C_F + \Delta C_F\right] + C_R + C_{AA}$$

Table IV: Suggested main particulars of the hull without and with PSS for CFD simulations

Characteristic		W/O PSS	With PSS	Units	Notes
Length between perpendiculars	L_{PP}	182.0	182.0	m	
Length of submerged body	L_{OS}	189.515	189.515	m	
Breadth	B	32.26	32.26	m	
Draught at aft perpendicular	T_{AP}	7.72	7.72	m	
Draught at fore perpendicular	T_{FP}	7.69	7.69	m	
Draught at midship	T_M	7.705	7.705	m	$T_M = \frac{T_{AP} + T_{FP}}{2}$
Static trim angle	t	-0.00907	-0.00907	deg	$t = arctan\left(\frac{T_{FP} - T_{AP}}{L_{PP}}\right)$
Displaced volume	∇	35820.3	35847.5	m^3	
Wetted surface area	A	7335.45	7360.93	m^2	
Longitudinal centre of buoyancy	x_{CB}	97.11	97.07	m	
Transversal centre of buoyancy	y_{CB}	0.00	0.00	m	
Vertical centre of buoyancy	z_{CB}	4.005	4.005	m	
Longitudinal centre of gravity	xCG	97.0643	97.062	m	Based on actual draughts
Transversal centre of gravity	yCG	0.00	0.00	m	
Vertical centre of gravity	zCG	7.705	7.705	m	Assumed to be T_M
Radius of gyration	k_{xx}	11.291	11.291	m	$0.35 \cdot B$
Radius of gyration	k_{yy}	45.5	45.5	m	$0.25 \cdot L_{PP}$
Radius of gyration	k_{zz}	45.5	45.5	m	$0.25 \cdot L_{PP}$

Table V: Suggested main particulars of the propeller

Characteristic		Value	Units	Notes
Propeller Diameter	D_P	5.8	m	
Pitch Ratio	$\frac{P}{D_P}$	0.723	-	
Area Ratio	$\frac{A_E}{A_0}$	0.631	-	
Hub diameter Ratio	$\frac{D_H}{D_P}$	0.15776	-	
No. of blades	Z	4	-	
Propeller Position	$Pos_{Prop}(x, y, z)$	3730.0, 0, 3210.0	mm	ship coordinate system

The total resistance is then calculated from

$$R_T = \frac{1}{2}\rho C_T S V^2$$

If we exclude the corrections for roughness, the total resistance R_T can be simply written as

$$R_T = R_{CFD} + R_{Air} + R_{BK}$$

R_{Air} is the air resistance, R_{BK} the resistance of the bilge keels. Table VI summarises the added air and bilge keels' resistance for each speed.

Table VI: Summary of added air and bilge keels resistance

Ship Speed	Air Resistance	Bilge Keels Resistance	Total Added Resistance
13.65 kn	18.23 kN	10.27 kN	28.04 kN
15.25 kn	22.75 kN	12.82 kN	35.57 kN
16.02 kn	25.11 kN	14.15 kN	39.25 kN
16.32 kn	26.06 kN	14.68 kN	40.74 kN

Comparing the submitted result for the difference between total resistance and propeller thrust shows that some of the calculations used different values for added resistance from the suggested values in the table.

For the CFD simulations, participants were free to use their preferred meshing methods, but the domain size was suggested to follow these values:

Minimum extent of domain = (-3.0, -2.0, -1.5) $\cdot L_{PP}$
Maximum extent of domain = (3.0, 2.0, 0.5) $\cdot L_{PP}$

The fluid properties, according to sea trial weather conditions, are defined as no wind, no waves, deep water, water temperature of 15 °C, water density of 1025 kg/m^3, and air density of 1.225 kg/m^3. The calculated kinematic viscosity of water from the temperature and salinity is $1.188 \cdot 10^{-6}$ m^2/s and for air is $1.457 \cdot 10^{-5}$ m^2/s.

Participants chose an adequate method for simulating the propeller thrust in the self-propulsion case. This can be fully resolved by the propeller or coupled RANS-BEM approaches. Simulations were performed at a given constant ship speed, while propeller RPM was adjusted to find the self-propulsion point. The submitted results presented in this paper are for the hull free to sink and trim, taking into account air resistance and bilge keels.

As for the surface roughness, the following values are used in the computations (the values are the same as for the JoRes1 tanker):

- the hull equivalent sand grain roughness k_s=53 μm (this corresponds to the measured Average Hull Roughness of 218 μm)
- the rudder equivalent sand grain roughness k_s=63 μm (this corresponds to the measured Average Rudder Roughness of 243 μm)
- the propeller equivalent sand grain roughness k_s=3.79 μm
- the PSS equivalent sand grain roughness k_s=6 μm

All calculations were performed in full scale and included the free surface.

5. Result of the Workshop

Seven participants (S01–S07) contributed simulation data for full-scale hull performance under free sinkage and trim conditions. Results were compared and analysed by RISE. Participants submitted computed torque, RPM, Resistance, thrust and power values for the JoRes5 bulk carrier operating at four speeds: 13.65, 15.25, 16.02, and 16.32 kn. Each case includes simulations with and without Pre-Swirl Stators (PSS). Some participants provided partial data, with missing entries for certain speeds or PSS configurations. Four of these submissions (S01, S04, S06 and S07) included all speeds for two cases (with and without PSS). While two participants (S02 & S05) only submitted one speed of 13.65 kn, one participant (S03) submitted all speeds for the hull without PSS. All participants used the RANS solver with $k\omega - SST$ turbulence model. Table VII lists the software type and the solver model of the submitted cases.

Table VII: CFD software used in the computations

Submission	Software	Propulsion Model
S01	FreSCo+	RANS_BEM Coupling
S02	StarCCM+ v23.10	Overset mesh for vessel, sliding mesh for propeller
S03	StarCCM+ v24.10	-
S04	SstarCCM+ v14.04	RBM model for propulsion
S05	StarCCM+ v23.02	-
S06	StarCCM+ v18.06	Sliding Mesh
S07	StarCCM+ v13.04	-

The computed power values show consistent trends across participants: Power increases with speed, as expected due to higher resistance and propulsion demand. PSS generally reduces power consumption, indicating improved propulsion efficiency. Table VIII summarizes the mean computed power across all valid submissions.

Table VIII: Average values of the submitted results of CFD simulations

Speed	Mean Power w/o PSS	Mean Power with PSS	Average Reduction
13.65 kn	5.0 MW	4.8 MW	3.0%
15.25 kn	7.2 MW	7.0 MW	2.7%
16.02 kn	8.9 MW	8.7 MW	2.1%
16.32 kn	9.6 MW	9.4 MW	2.0%

The submitted results included values for trim angle, sinkage of the hull, thrust exerted, torque and RPM of the propeller and hull resistance. Fig.7 presents the submitted values for the sinkage and trim of the vessel at different speeds for the cases with and without PSS.

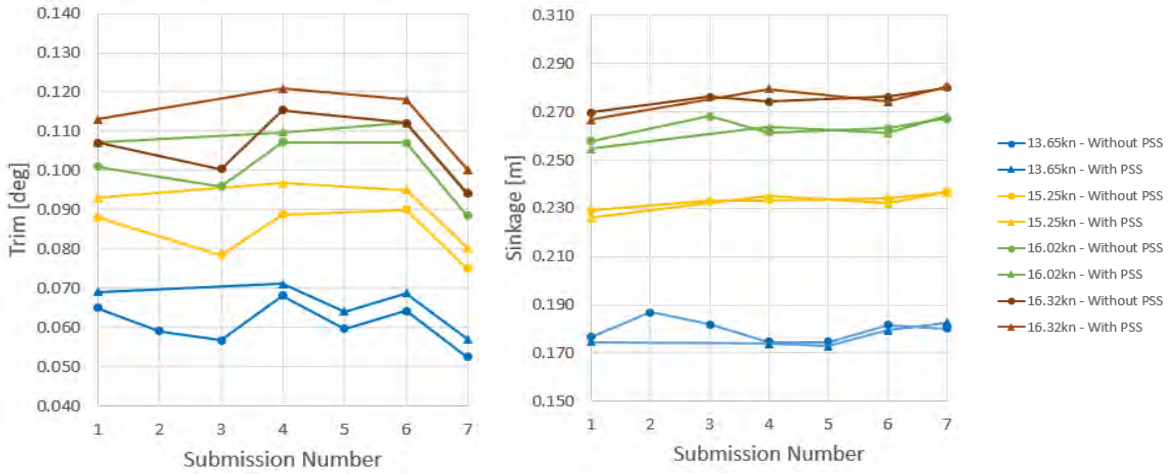


Fig.7: Computed Trim angle and Sinkage at different speeds

The power reduction due to PSS varies slightly among participants, ranging from ~1% to ~5.7%, depending on speed. Comparing the thrust and resistance forces showed that S06 did not consider the added resistance due to air and appendages, and S01 overestimated the added resistance at a lower speed of 13.65 kn.

Fig.8 presents the calculated power at different speeds for two cases. The percentages of power reduction due to PSS are also presented, and the horizontal solid line indicates the sea-trial result for the given ship speed.

Fig.9 shows the computed propeller rotational speeds, while horizontal lines represent sea trial values for the given ship speed and case. Fig.10 shows the computed thrust force at different ship speeds and cases.

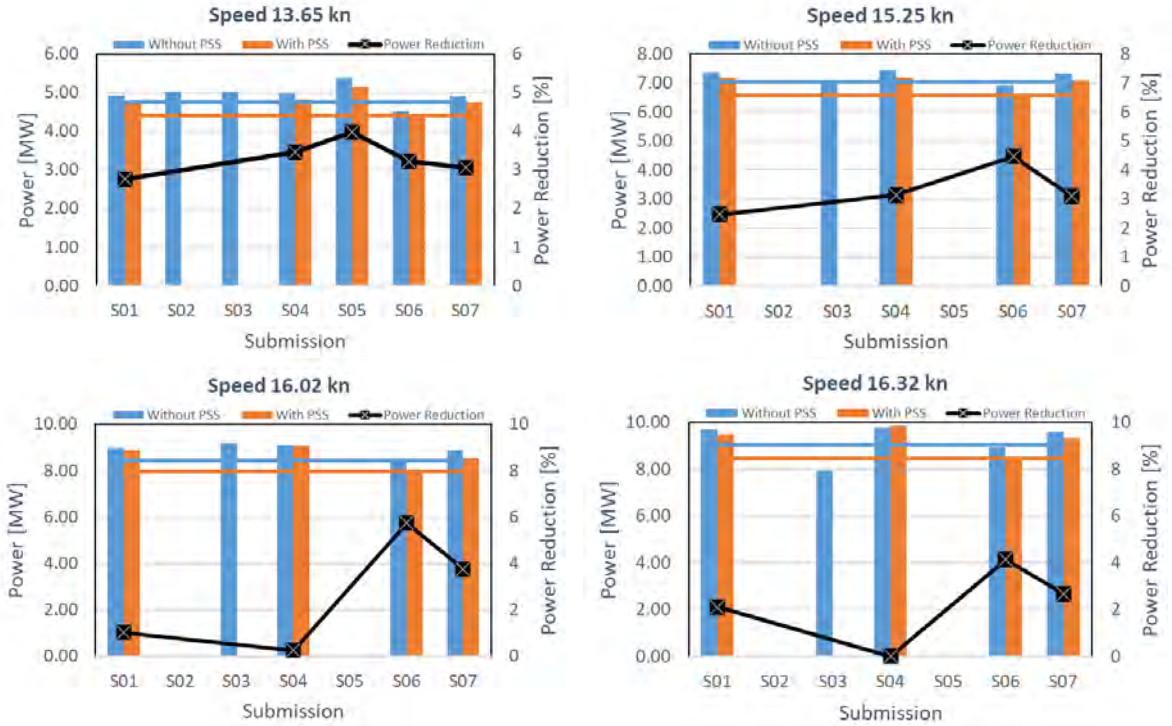


Fig.8: Power calculated at different speeds for two hulls with and without PSS. Horizontal lines represent sea trial values.

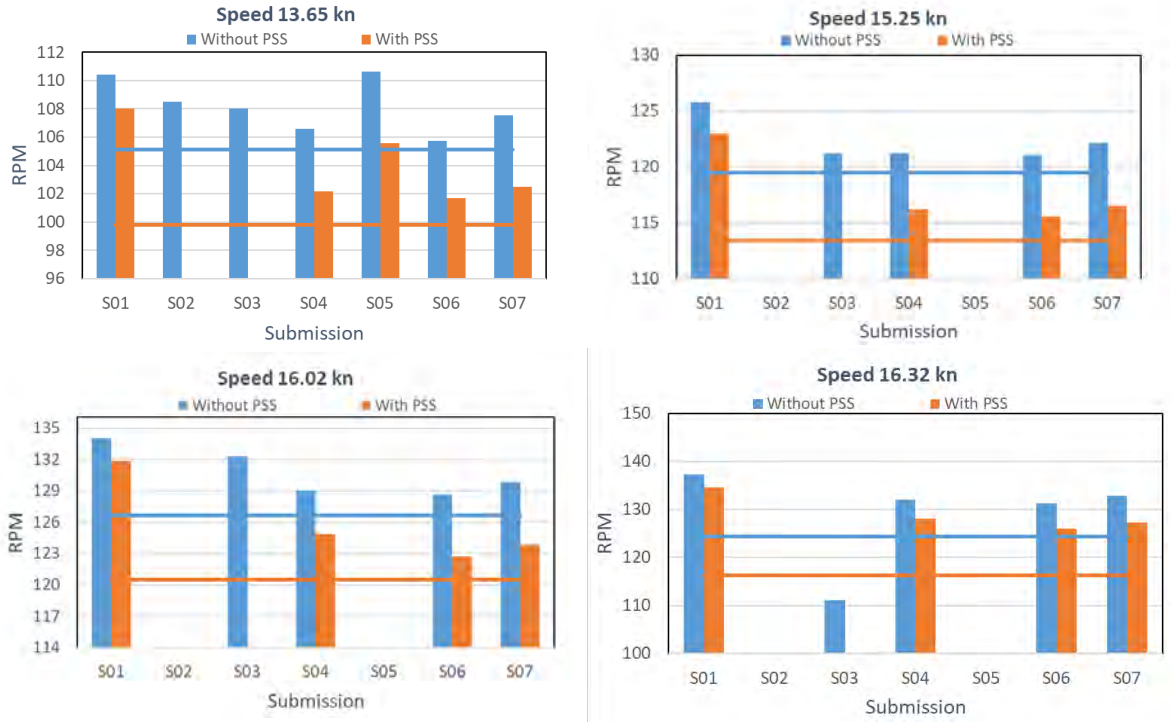


Fig.9: Computed propeller rotational speeds. Horizontal lines represent sea trial values

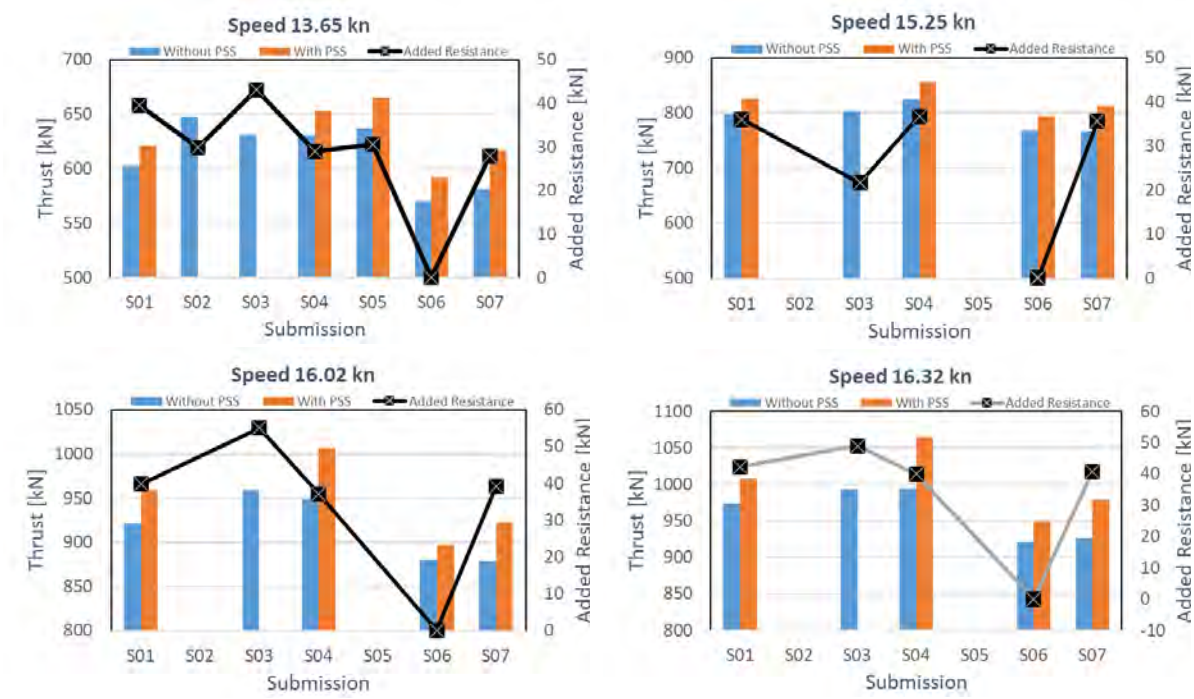


Fig.10: Computed thrust at different speeds

6. Summary and Conclusion

The average submitted data across all participants overestimated the RPM and power values according to Table IX.

Table IX: Difference between RPM and Power valued computed and sea trial

Speed	Without PSS		With PSS	
	RPM	Power	RPM	Power
13.65 kn	2.94%	4.41%	4.19%	8.15%
15.25 kn	2.32%	2.04%	3.92%	6.59%
16.02 kn	3.29%	5.56%	4.41%	8.25%
16.32 kn	-0.40%	1.74%	4.85%	9.79%

The average submitted power saving of the PSS is under-estimated by 50% according to Table X.

Table X: Power saving due to PSS

Delivered Power Reduction		
Speed	Computed	Sea Trial
13.65 kn	3.29%	7.10%
15.25 kn	3.30%	6.65%
16.02 kn	2.69%	5.79%
16.32 kn	2.00%	6.08%

Unfortunately, despite the commitment of the participants and their submissions, the results are not fully representative because two out of seven participants did not submit the complete set of results, and one participant did not take into account the added resistance due to superstructure and bilge keels. So, only 4 out of 7 sets can be compared. Moreover, S04's results do not appear very consistent; they predicted that at a speed of 16.32kn, the vessel would require more power with PSS rather than without

PSS. All other predictions do not show the exact same results, but they are, in general, consistent with the measurements. *Andersson et al. (2022)* introduced a similar concept: an energy-saving duct was built for KVLCC, and participants were asked to run the case with and without the duct. The results of that workshop were very inconsistent, and the saving prediction varies from −2.9% to +3.4% excluding several outliers with ±10%. Therefore, compared to that case, there is an improvement in the results consistency. As the data (geometries, sea trials results, etc.) is now fully available in the public domain at the JoRes website, it is expected that other researchers will perform further simulations, and the results dataset will become larger and more representative for comparison.

References

ANDERSSON, J; SHIRI, A.; BENSOW, R.; YIXING, J.; CHENGSHENG W.; TURNOCK, S.; WERNER, S. (2022), *Ship-scale CFD benchmark study of a pre-swirl duct on KVLCC2*, J. Applied Ocean Research 123

ITTC (2008), *Recommended Procedures and Guidelines, Performance, Propulsion 1978 ITTC Performance prediction method*, Int. Towing Tank Conf.

An Integrated Real-Time Ship Operation Optimisation System to Improve Energy Efficiency and Schedule Reliability of Shipping Navigation and Port Calls

Sergio Ribeiro e Silva, University of Lisbon, Lisbon/Portugal, ribeiro.e.silva@tecnico.ulisboa.pt
Miguel Bento Moreira, University of Lisbon, Lisbon/Portugal, bento.moreira@tecnico.ulisboa.pt

Abstract

A techno-economic analysis of an integrated real-time Ship Operation Optimisation System (SOOS) to reduce fuel consumption and emissions from shipping navigation and port calls has been conducted to improve energy efficiency and simultaneously turn a case-study vessel compliant with CII proposed by IMO. This new robust integrated real-time digital solution involves a significant number of both technical and operational measures "in practice" aiming to optimise operational efficiency (during navigation and port calls). Namely, the tool will be capable of situational awareness and decision support to reduce fuel consumption and GHG emissions from shipping and must be combined with intrinsic vessel systems to improve vessel hydrodynamic performance, resulting also in improved vessel safety and widening of the operational weather window. In particular, use of CFD simulations have been conducted with the aim of obtaining reliable predictions of calm water resistance and trim optimisation calculations at a given ship loading condition, which combined with corrections for current, wind and wave effects lead to the development of this optimisation-based approach to reduce fuel consumption and GHG emissions from shipping navigation. Furthermore, development of a voyage planning module based on weather routing to save fuel and increase safety and schedule reliability (in terms of 'Just in Time' arrival to port) has been envisaged. Finally, before integrating the voyage planning module with the other SOOS modules and conducting full-scale demonstrations during sea trials, all components must undergo rigorous testing in a virtual environment to properly de-risk this new technology.

1. Introduction

As the most energy efficient mode of transportation, the maritime transport sector is one of the major sectors of cargo shipping of goods around the world, but is also responsible for 681 tonnes (t) of CO_2 every year, *Hieminga and Luman (2023)*. Hence, IMO is targeting a drastic reduction of Green House Gas (GHG) and CO_2 for shipping, *IMO (2023)*. These ambitious targets involve major changes in the way ship owners, in general, and the maritime industry sector operates, bringing the need to adopt new technologies, investing in greener alternative fuels and adopting practical measures to improve current energy efficiency of the means of transportation.

In addition to this most challenging context of the maritime industry sector, many weather routing service providers claim the ability to save fuel and increase safety and schedule reliability. However, many seamen's lives are frequently put at risk since more than 3,000 containers are lost overboard every year. According to *WSC (2023*), the average annual loss for the two-year period 2020-2021 saw an increase to 3,113 from the 779 of the previous period, driven by major incidents. In 2020 the ONE Opus lost more than 1,800 containers in severe weather. The Maersk Essen also experienced severe weather in 2021 that resulted in the loss of some 750 containers. Also, a study conducted by *Gershanik (2011)* revealed that weather routing helped reduce ship damages from rough weather by 73%, maintenance costs by 29%, and cargo damage lawsuits by 87%. At the same time, ship delays from unfavourable weather were reduced by 80%; fuel savings amounted to about 6%. With exaggerated capabilities and unsubstantiated benefits being advertised by weather routing companies, port authorities, ship owners, operators or charterers often face the difficult task of selecting the right service provider and level of technology suitable for their operations.

Anticipating that fuel prices in years to come will remain high due to war in Ukraine, *IEA (2023)* and the conflict in the Middle-East plus the recent emphasis on reducing GHG emission in Europe, have resulted in renewed interest in further optimising ship performance. A recent DnV study, *DNV (2022)*

indicated that while hydrodynamic performance (hull coating, hull form and trim optimisation and regular propeller cleaning) can achieve 5 to 15% reduction in fuel consumption and associated GHG emissions, more than 20% improvement can be achieved through technical and operational measures such as speed management, fleet planning and weather routing (the so-called logistics and digitalisation). Moreover, an experimental campaign conducted by HSVA with scaled models of a containership have demonstrated that hydrodynamic performance can be further improved in case an anti-rolling tank is installed to reduce ship motion waves, *HSVA (2020)*.

To tackle the problem above, firstly, a numerical program has been developed at University of Lisbon, IST to evaluate added resistance in waves using output data of a standard strip theory seakeeping program developed by *Ribeiro e Silva (2008)*. In this case, the strip theory code is based on Frank's Close-Fit method and the added resistance in waves is evaluated using the formulation originally proposed by *Salvesen (1978)*. The program has provided good results against experimental data available in literature, especially for slender ship forms, *Ribeiro e Silva (2011)*. The numerical predictions presented in here have been compared against experimental data relative to ship's models with the same *L/B* ratio. Additionally, another Computational Fluid Dynamics (CFD) solver of the Navier-Stokes (N-S) type has been utilised to provide additional numerical predictions of the ship's total resistance in calm water and more detailed information on the flow characteristics around the ship's hull for distinct trim angles, further details in *Ribeiro e Silva (2024a*). More recently progressive increase in memory and speed of computers favours the utilisation of CFD N-S solvers, for current voyage planning purposes, i.e., simulations close to real-time for variable metocean (meteorological and oceanographic) conditions, in conjunction with Salvesen method to predict added resistance in waves seems to be the most suitable decision support tool. Where a multi-dimensional (velocity and heading), multi-disciplinary constrained (rms ship motions and eventually with prevention of dynamic instabilities in waves), single objective (fuel consumption) optimisation algorithm has been proposed in order to take into account not only the pertinent fuel savings, but also the safety aspects of the voyage. Efforts have been focused so far on key technical-economic challenges that can demonstrate cost effectiveness and applicability of the concept. In particular, use of CFD simulations have been conducted by *Ribeiro e Silva (2024b)* with the aim of obtaining reliable predictions of calm water resistance, which combined with corrections for current, wind and wave effects lead to the development of this optimisation-based approach to reduce fuel consumption and emissions from shipping navigation.

Furthermore, development of a voyage planning module based on weather routing to save fuel and increase safety and schedule reliability (in terms of Just in Time arrival to port) has been envisaged. Hence, prior to integration of the voyage planning module with the other Ship Operation Optimisation System (SOOS) modules and their full-scale demonstration during sea trials, all these modules must be extensively tested in a virtual environment to properly de-risk this new technology.

Firstly, the methodology used in this paper is presented in the Theoretical Background section, where an optimisation-based approach for enhanced fuel efficiency and safety aboard is described.

Secondly, in the Numerical Results section, some preliminary figures on the performance of the newly developed voyage planner are shown for a synthetic environment to demonstrate the capabilities of the tool. Next, the performance was assessed using weather forecast predictions from MOHID, *Neves (1988)*, i.e., a scenario closer to the real world that takes into consideration all the most relevant metocean conditions. Namely, the range of surface currents, wind loading conditions and sea states which a typical containership usually operates in the Atlantic West coast of Portugal were simulated in order to set a numerical model that could be utilised to calculate the optimised fuel consumption for a desired average speed between two ports.

As mentioned in Conclusions, it is believed that calculation of specific hydrodynamic responses such as added resistance in waves for a real-time loading condition of the vessel. This represents a major advantage over other commercially available tools, as it also allows the designer to define the most suitable hull form and superstructure area for the most energy efficient mode of operation of the vessel.

2. Theoretical Background

In general, the hydrodynamic performance of a ship is influenced by the surrounding environmental conditions. In this context the existing space and time realizations of wind, waves and ocean current conditions can be defined as the environmental factors in any voyage. These factors will affect the fuel consumption by changing the power requirements for the propulsion of the vessel. Hence, this section attempts to presents a summary of the basic concepts in the realm of power estimates for the ship propulsion.

2.1. Power Curve Estimation

Ship's resistance is particularly influenced by the ship's hull form, speed, displacement and trim. In addition to these calm water parameters, added resistance in waves or even wave-induced roll motion can also have a significant impact on the power delivered by the propeller. The total resistance R_T consists of several resistance components acting on the ship, which will be briefly described in this section, while also referring to some relevant works that provide a more thorough and analytical presentation of methodologies to estimate those components of the ship's resistance.

Hydrodynamic analysis for a specific hull form and ship loading condition can be used to calculate the components resistances acting on the ship by means of dimensionless resistance coefficients, for example, *Harvald (1983)*. This analysis may consist of using towing tank tests with scaled models, developing CFD models to simulate the flow around the ship, quasi-experimental methods using results of experiments and calculations. In case, calm water resistance is obtained, then seakeeping models can be utilised to estimate ship motions in irregular waves.

In this study, CFD Simerics-MP software considering a marine template to calculate ship resistance is used to determine calm water effective hull resistance at a given loading condition and range of ship speeds, i.e., the so-called power curve in calm water. However, calm water is very seldom encountered in real world conditions, particularly in ocean going voyages. For example, in the North Atlantic the probability of encountering calm water conditions is only 26 days in a year, i.e., 0.7%.

According to *Scheekluth and Bertam (1998)*, calm water total resistance of a ship is made up of a number of different components, which are caused by a variety of hydrodynamic factors, and interact one with each other in an extremely complex way. Adopting a reductionist approach, calm water resistance, R_T , consists of a viscous resistance plus the resistance due to the Kelvin waves generated by the hull. Hence, the total resistance coefficient, $C_T = C_V(Rn) + C_R(Fn)$, can be obtained from:

$$R_T = \frac{1}{2}\rho_{SW} S_W V_S^2 C_T = \frac{1}{2}\rho_{SW} S_W V_S^2 (C_V + C_R), \quad (1)$$

ρ_{sw} is the density of seawater, V_s the ship's speed over water and S_w the total wetted area of the hull. Note that Eq.(1) is a simplified formulation of the complex nature of ship resistance in calm water, where viscous resistance coefficient, C_V, is assumed to be only dependent of Reynolds number Rn, whereas the coefficient of residuary resistance, C_R, mainly composed by wavemaking resistance, C_W, is assumed to be dependent on Froude number Fn.

Once the main resistances have been estimated, one can calculate the required effective power with appendages, P_{EA}, to move the ship through the water at the required sailing speed. Based on that, the required nominal power at the main engine shaft P_S can be calculated using the shaft-line efficiencies:

$$P_S = \frac{P_{EA}}{PC} = \frac{R_{EA} V_S}{QPC\eta_S} = \frac{(R_T + R_{APP})V_S}{\eta_H \eta_O \eta_R \eta_S} \quad (2)$$

The behaviour of the flow around the hull determines the ship resistance and the wake at the propeller, which are interrelated, *Ribeiro e Silva and Bento Moreira (2024a)*. While the magnitude of the resistance

directly determines the power requirements, the magnitude of the wake and its distribution at the propeller plane also affects the power requirements as well as the performance characteristics. When the wake distribution is highly non-uniform at the propeller plane, this is a major source of poor propulsive efficiency, cavitation, vibration and noise. Assuming some simplifications, the main engine that satisfies the propeller's demands, and allows the ship to sail at its nominal speed during voyage, can be selected.

2.2. Wind Resistance

In the first place, it is recalled that wind is the cause of the creation of waves so that their incoming direction is sometimes practically the same, and simultaneously wind acts as a force on the vessel known as wind resistance. Wind resistance will affect all surfaces of the ship above the sea-surface as well as cargo when the latter is above the hull, as it is the case of most containerships. According to Bernoulli's equation, wind resistance either in the *x* or *y* direction is directly proportional to the projected frontal A_F or lateral A_L areas of the ship above the waterline, the density of the air, ρ_{air}, and the square of the wind speed. These two projected areas are defined by the above-water part of the main hull and any superstructures (e.g., cargo, bridge, funnel, and equipment). Normally wind represents around 2% of the total resistance, with the notable exception of containerships where due to both large projected frontal and lateral areas of the vessel (due to 60% of containers on-board are piled-up above the main deck) the contribution can reach up to 10%.

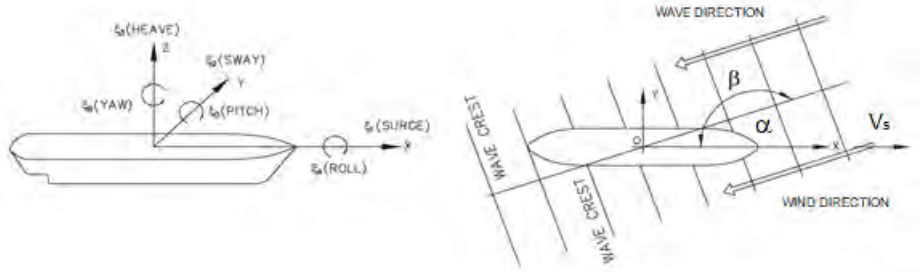


Fig.1: The non-inertial body-fixed tri-axial coordinate system of the ship, illustrating the six modes of motion and the definition of the true wind vector (α) and wave (β) incoming direction angles

The wind velocity is typically decomposed into two components: the fair wind, which is what the ship is facing during sailing (same speed but opposite direction of the vessel), and the true wind, which is the actual wind speed and direction at the current position at sea. The wind resistance can be computed using either a semi-empirical formulation originally proposed by *Gould (1982)* or a standard CFD tool Simerics-MP and is essentially the wind that the vessel would face due to the fact it is sailing, while the latter would be the wind the vessel would face at the same location if she was anchored. Adding these two vector components, the resulting apparent wind vector can then be used to calculate the total wind resistance. The true wind induced longitudinal F_{wx} and transverse force F_{wy} components and the yawing moment M_w , Fig.1, can be calculated using the following equations adopted by *Isherwood (1972)*, *Gould (1982)* and *Blendermann (1994)*:

$$\begin{bmatrix} F_{wx}(\alpha_w) \\ F_{wy}(\alpha_w) \\ M_w(\alpha_w) \end{bmatrix} = \frac{1}{2}\rho_{air} {V_w}^2 \begin{bmatrix} C_{wx}(\alpha_w)\, A_F \\ C_{wy}(\alpha_w)\, A_L \\ C_w(\alpha_w)\, A_L\, L_{OA} \end{bmatrix}, \tag{3}$$

ρ_{air} and V_w stand respectively for the air density and true wind speed and $C_{wx}(\alpha_w)$, $C_{wy}(\alpha_w)$ *and* $C_w(\alpha_w)$ represent the non-dimensional wind load coefficients dependent on the given true wind angle of attack α_w. $C_{wx}(\alpha_w)$ is given with respect to the frontal projected area of the ship, A_F, whereas $C_{wy}(\alpha_w)$ is given with respect to the lateral projected area of the ship, A_L. The yawing-moment coefficient is given with respect to the product of the lateral projected area of the ship by the ship's length overall, L_{OA}.

Considering $C_{wx}(0)$ and $C_{wy}(0)$, from Eqn. (3) one can obtain $F_{wx}(0°)$ and $F_{wy}(90°)$ and the true wind

induced force F_w , at any given angle of attack α_w can be computed as:

$$F_w(\alpha_w) = F_{wy}(90°)\left(\frac{2sin^2(\alpha_w)}{1+sin^2(\alpha_w)}\right) + F_{wx}(0°)\left(\frac{2cos^2(\alpha_w)}{1+cos^2(\alpha_w)}\right) \quad (4)$$

Finally, the true wind induced force components can be calculated as:

$$F_{wx}(\alpha_w) = F_w(\alpha_w)cos(\alpha_w) \quad (5)$$

and

$$F_{wy}(\alpha_w) = F_w(\alpha_w)sin(\alpha_w). \quad (6)$$

2.3. Added Resistance in Waves

Both involuntary and voluntary speed reductions are taken into account to avoid over-predicted ship speed and wrong diversion decisions when facing rough weather, not to mention inaccurate estimates of fuel consumption and time of arrival. To prevent this problem, added resistance in waves for a specific loading condition and a given hull form have been computed as well using a state-of-the art numerical tool. Note that this additional component of resistance called added resistance in waves, R_{aw}, is heavily non-linear. For the calculation of the added resistance in waves, *Salvesen (1978)* introduced the added resistance in waves, given by:

$$R_{aw} = -\frac{1}{2}k\ cos\ cos\ \beta\ \sum_{j=3,5} \quad \xi_j\left\{\left(F_j^I\right)^* + \hat{F}_j^D\right\} + R_7, \quad (7)$$

k is the wave number, $\left(F_j^I\right)^*$ the Froude-Krylov force and moment, $\hat{F}_j^D$ the diffraction force and moment, R_7 the added resistance due to diffraction potential, ξ_j the ship's displacement induced by waves in j direction. For head waves, the equation can be expressed in the non-dimensional format as, *Ribeiro e Silva and Bento Moreira (2024a)*:

$$R_{aw}{}' = \frac{\hat{F}_3+\hat{F}_5+R_7}{\rho_{SW}g\left(\frac{B^2}{L_{PP}}\right)(\zeta_\omega^a)^2} \quad (8)$$

Next, a practical simplification is introduced to estimate added resistance in waves directly from head sea result, where added resistance in waves R_{aw} at any angle β relatively to head waves is given by:

$$R_{aw}(H_S, T_P, \beta) = R_{aw_{180^o}}(T_P)\left(\frac{H_S}{2}\right)\beta \quad (9)$$

H_S and T_P stands respectively for the significant wave height and mean wave period. cos²β is a spreading function of the relative incoming wave direction hitting the ship at angle β.

2.4. Added Resistance due to Wave Induced Roll Motion

Most merchant vessels (containerships, Ro-Ro, bulk-carriers and tankers) have low levels of roll stabilisation attained via hull appendages (e.g., bilge-keels). Due to comfort operational requirements, passenger vessels usually have a higher level of roll stabilisation usually attained with active fins in congress with a pair of bilge-keels. As demonstrated in the experimental studies conducted at *HSVA (2020)*, whereas seakeeping characteristics can be significantly improved with these two types of roll stabilisation system, in case elimination of additional towing power and propeller power is deemed as desirable, then installation of anti-roll tanks (without additional hull appendages) represents a much more viable solution. Considering the impact anti-roll tanks will have in the reduction of fuel consumption and improvement of seakeeping characteristics, where significant fuel savings per day at all speed ranges can be easily obtained, a ship fitted with either a passive or controlled U-type tank fitted with air-valves along with a trim and a voyage optimisation systems represent the best holistic approach to ship operation optimisation.

Similarly to added resistance in waves, roll motion adverse effect on delivered power to the propeller must be determined. The roll responses in waves are based on Frank's Close-Fit method from which the roll Response Amplitude Operators (*RAO*) for different headings and speeds could be computed.

According to *Ribeiro e Silva et al. (2006)* and *Ribeiro e Silva (2008),* after some full-scale validation trials conducted by *Ribeiro e Silva et al. (2005)* enhanced capabilities have been added to the frequency domain computer simulation codes in order to model and assess stabilised ship responses fitted with bilge keels, passive U-type tanks or active fins. According to *Ribeiro e Silva et al. (2022)* U-tank scaled model tests on a moving platform were conducted at CEHIPAR to validate numerical models and compare and contrast the performance of anti-roll tanks controlled with air valves against those of the passive type. More recently, the Gyroscopic Roll Stabilisation has been implemented as well by *Ribeiro e Silva and Varela (2022)*, and the hydrodynamic problem has been further generalised for less slender hull-forms using a panel method for the analysis of the interaction between the vessel and the surface waves with speed corrections instead of a standard strip-theory method.

In general, the six DoF fully coupled governing roll motion equations for a ship fitted with a roll stabilisation system is given by Eqn. (10).

$$\left(M_{kj} + A_{kj}\right)\ddot{\xi}_j + B_{kj}\,\dot{\xi}_j + C_{kj}\xi_j = F_k^W + F_k^{Stab} \text{ for } k, j = 1,\ldots,6 \tag{10}$$

2.4.1. U-type Tank Stigter's and Lloyd's Numerical Models

Lloyd (1989) introduced some simplifications on *Stigter's (1966)* original formulation by assuming a U-tank having two lateral reservoirs whose walls are parallel and perpendicular to the connecting channel. On the other hand, the linear model proposed by Lloyd represented a further generalisation of Stigter's pioneering studies by extending the formulation to four degree-of-freedom: sway, roll, yaw and the dynamics of the tank.

When the ship is rolling in regular beam waves both quantities φ - ship rolling angle, and τ - water free-surface angle inside the U-tank vary in time with a frequency equal to that of the harmonic wave motion. This simplified U-tank problem, only two degrees-of-freedom are being considered and the following governing equations can be derived:

$$\begin{bmatrix} I_{xx} + a_{44} & -a_{4\tau} \\ a_{\tau 4} & a_{\tau\tau} \end{bmatrix} \begin{Bmatrix} \ddot{\phi} \\ \ddot{\tau} \end{Bmatrix} + \begin{bmatrix} b_{44} & 0 \\ 0 & b_{\tau\tau} \end{bmatrix} \begin{Bmatrix} \dot{\phi} \\ \dot{\tau} \end{Bmatrix} + \begin{bmatrix} c_{44} & -c_{4\tau} \\ c_{\tau 4} & c_{\tau\tau} \end{bmatrix} \begin{Bmatrix} \phi \\ \tau \end{Bmatrix} = \begin{Bmatrix} F_4^W \\ 0 \end{Bmatrix} \tag{11}$$

where all the coefficients can be found in *Ribeiro e Silva et al. (2006).*

2.4.2. HSVA Model Testing on Increased Ship Resistance due to Rolling

Considering the impact anti-roll tanks will have in the reduction of fuel consumption, Table 15 of *HSVA (2020),* it can be inferred that roll motions are detrimental to seakeeping and also increase ship resistance and thus fuel consumption. In particular, it can be inferred from this particular table of HSVA report that fuel savings of 6 t/day at 14 kn or 10.4 t/day at 21 kn can be easily obtained for a generic containership fitted with a standard free-surface anti-roll tank.

A practical simplification has been introduced to estimate additional power due to roll motion in waves, *Ribeiro Silva and Bento Moreira (2024b),* where the additional power to bare hull effective power in still water ratio is simply given by:

$$\frac{\delta P_E}{P_{E\,Barehull}}(H_S, T_P, V_S, \beta) = \left[p_1\,\phi_{sig}^2(H_S, T_P, V_S, \beta = 90^o) + p_2\,\phi_{sig}^3(H_S, T_P, V_S, \beta = 90^o)\right] sin^2\beta, \tag{12}$$

$sin^2\beta$ is another spreading function of the relative incoming wave direction hitting the ship at angle β

relatively to the loss of velocity due to roll motion in beam waves, and whose limits of application are between 0 and π (head and stern waves). This function will vary with the relative incoming wave direction and the ship roll response in waves, which can then be transformed into a loss of velocity due to roll motion in waves, $V_{S\,roll}$, using experimental measurements of increased effective power, $P_{E\,roll} = R_{E\,roll} V_{S\,roll}$, of a scaled model of a containership due to beam waves induced roll motion (HSVA report). Notice that using local spectral peak period and ship's Froude number, the encountered wave period can be determined, from which the roll motion amplitude can be calculated using both roll *RAO* and the wave spectrum. Finally, using the polynomial fit of the experimental results of P_E versus φ_{sig} (in degrees), previously obtained from model testing at HSVA, i.e., $p_1 = 112.332 \times 10^{-3}$and $p_2 = 3.518 \times 10^{3}$, the full-scale ship's power loss (or velocity loss, as shown in Eqn. 12)) can be estimated for containerships having similar hullforms.

2.5. Trim Effect on Ship Resistance

CFD with RANS turbulence models can address a wide variety of flows including external flows around bodies of a certain shape, i.e., statistically steady flows that require streamlined shapes aligned with the incoming flow where boundary-layers do not exhibit significant flow separation. However, for trim optimisation these simulations must be supported by relevant background experience in the realm of CFD tools utilisation and must be subjected to a dedicated Verification and Validation (VandV) procedure.

CFD simulations of ship performance in waves were mainly built around an advanced actuator disk model implemented either in or Simerics-MP, *Ribeiro e Silva and Bento Moreira (2024a)*. which reads the open water performance of a real propeller. Next, considering the geometry of the hull, accurate ideal trim predictions of the total CFD resistance in calm water were determined for enhanced stability and fuel consumption optimisation purposes by altering the vessel's position of centre of gravity of the ship, *CG(x,y,z)*. The instantaneous location of the centre of gravity of the ship, *CG(x,y,z,t)* depends on magnitude and location of a large number of discrete weights aboard, including fuel and ballast water in the aft and peak tanks, so that the required instantaneous trim angle variation for enhanced stability and fuel consumption optimisation purposes can be easily calculated by SOOS.

Finally, in case non-stationary simulations are deemed as desired, added resistance in waves component as well as wave induced roll motion component of total ship's resistance can be computed using SOOS for any particular loading condition.

3. Weather Routing Optimization

3.1. Objective function

The total fuel consumption *C* and the total travel *T* time between points *A* to *B*, depends on the route *L* chosen and can be computed using the following line integral with respect to the arc length:

$$C(L) = \int_L \frac{\frac{\partial M}{\partial t}}{V} ds \quad (13)$$

and

$$T(L) = \int_L \frac{1}{V} ds \quad (14)$$

where $\frac{\partial M}{\partial t}$ and *V* stands respectively for fuel consumption time rate and the speed along the chosen route *L*. Note that $\frac{\partial M}{\partial t}$ and *V* depends particularly on environmental conditions such as wind, currents and waves. Therefore, the route that minimizes total fuel consumption *C* or the total travel time *T*, in such conditions, is not the straight (or geodesic) path from *A* to *B*. Although the length of a route has a similar

influence on both total fuel consumption and total travel time, the routes that minimize each of these parameters *C* or *T* can be very different. Therefore, a simple optimisation of total consumption may not result in a travel time that is compatible with the trip's objectives and in particular with the "just in time" concept.

In *Ribeiro e Silva and Bento Moreira (2024a,b)*, the selection of optimal paths *L* considered only the minimization of total fuel consumption *C*. Here, we test a performance parameter *F*, which accounts for the importance given to both total fuel consumption *C* and total travel time *T*. Let C_{dir} and T_{dir} be the total fuel consumption and the passage time, respectively, for the direct rhumb line route L_{dir} from point *A* (the starting point) to point *B* (the destination). Define *F* (Fitness) as the following weighted average

$$F(L) = \frac{\alpha}{100}\frac{C(L)}{C_{dir}} + \frac{\beta}{100}\frac{T(L)}{T_{dir}} \quad , \tag{15}$$

$0 \leq \alpha \leq 100\ \%$, $0 \leq \beta \leq 100\ \%$ and $\alpha + \beta = 100\ \%$. Minimizing *F(L)* with α = 100% is equivalent to minimizing *C*(*L*), and minimizing *F*(*L*) with β = 100 % is equivalent to minimizing *T*(*L*). The relative importance of *C* and *T* can be weighted by appropriately adjusting the values of α and β. The use of the parameter *F* allows for assigning the desired importance to *C* and *T* in the minimization process that leads to the selection of the path *L*.

3.4. Optimisation algorithm

The approach that has been adopted to find the route *L* that minimizes the global fuel consumption is the Vectorised Simulated Annealing (VSA) technique which is based on the Simulated Annealing (SA) method, *Press (2007).* As it can be seen *in Ribeiro e Silva and Bento Moreira (2024a)*, this method can be roughly described using the analogy with the metallurgical process of annealing to bring a metal from an high energy/temperature state to a crystal lattice state of minimum/temperature energy, where a slow and gradual decrease in temperature will allow the system to properly explore the search space and assume the configuration of minimum energy by the end of the annealing process. SA method was developed by Metropolis and co-workers in 1953, *Metropolis et al. (1953).* In the simulated annealing optimization method, the procedure for obtaining an optimal system configuration involves imposing perturbations on an initial system configuration. The amplitude/intensity of these perturbations (temperature) decreases over a succession of phases. In each phase, a perturbed configuration is chosen to serve as the starting point for the next perturbation of lower amplitude/intensity. At each phase, the selection of the chosen configuration must obey the following criterion: it should be better than the configuration chosen in the previous phase (high probability) or it may be accepted (low probability) that it could present a worse performance. The strategy of decreasing the "temperature" and adjusting the probability of accepting configurations with worse performance during the "cooling" process are the key to avoiding the system's evolution toward local optima, achieving an effective exploration of the search space, and ensuring the convergence of the system's configurations to an optimal configuration. A comprehensive description of this method, also known as the Metropolis-Hastings algorithm, can be found in *Hitchcock (2003)*. The VSA method applies components of the SA algorithm in parallel to a population of systems that constitute the components of a vector. This methodology has been successfully applied in previous works such as *Mauricio and Bento Moreira (2023).* In the application of the VSA method, paths will be modelled by a process of proper concatenating oriented rhumb lines. Note that any path between *A* and *B* can be arbitrarily approximated in the aforementioned manner. So, define *N* concatenated oriented segments L_i (rhumb line legs) that establish a route *L* between points *A* and *B*. This path is thus defined by *N*-1 yaw points P_i . Then, total fuel consumption *C* and the total travel time *T,* reads:

$$C(L) = \sum_{i=1}^{N} \quad C(L_i) \tag{16}$$

$$T(L) = \sum_{i=1}^{N} \quad T(L_i), \tag{17}$$

$C(L_i)$ and $T(L_i)$ is the fuel consumption and the passage time in leg L_i, respectively.

Considering Eq.(15), the global fitness F associated to a given route L will be:

$$F(L) = \sum_{i=1}^{N} \quad F(L_i), \tag{18}$$

$C(L_i)$ and $T(L_i)$ can be numerically integrated (with respect to the arc length) using a standard numerical integration method (e.g., trapezoidal rule) and adopting an appropriate spatial discretization of the leg over the corresponding leg L_i.

The aim is to determine the positions of the N-1 yaw points P_i thus determining the path L that minimizes the global fitness F.

As previously mentioned, fuel consumption rate $\frac{\partial M}{\partial t} = \frac{\partial M}{\partial t}(x, y)$ and speed $V=V(x,y)$ along the routes depend on environmental conditions at the corresponding spatial coordinates (x,y).

We postulate that the trajectory from point A to B is carried at a convenient desired the ship's speed over water V_0 (desired/base speed) corresponding to a calm water fuel consumption rate of $(\frac{\partial M}{\partial t})_0$ and corresponding to a nominal power at the main engine shaft P_0 .
In the estimation of the effective fuel consumption time rate $\frac{\partial M}{\partial t}$ we will consider the following environmental effects:

1. The wind resistance depending on the wind intensity and wind direction;
2. The added resistance in waves depending on the significant wave height and mean wave period;
3. The added resistance due to wave induced roll motion depending on the significant wave height and mean wave period;
4. The effect of drift from the athwartships component of the current.

In the estimation of the effective ship's speed V, only the effect of the current component in the direction of the heading shall be considered.

The power curve P [kW] versus speed V [ms^{-1}] is modeled by means of a 3rd order polynomial:

$$P = a_1 V^2 + a_2 V^3 \tag{19}$$

$a_1 = -0.275$ and $a_2 = 6.510$ following *Ribeiro e Silva and Bento Moreira (2024a,b).*

The resistance curve R [kN] versus speed V [m/s] can be deduced from (19) reading:

$$R = a_1 V \quad + a_2 V^2 \tag{20}$$

The fuel consumption rate $\frac{\partial M}{\partial t}$ [kg/s] versus speed V [m/s] can be deduced from (19) considering MCR = 182.5 [g/kWh]:

$$\frac{\partial M}{\partial t} = k(a_1 V^2 + a_2 V^3) \tag{21}$$

where $k=\frac{0.1825}{3600}$.

So, *Ribeiro e Silva and Bento Moreira (2024a)*, in the absence of wind, waves, and currents, let P be the power required to maintain speed V in [m/s], let R in [kN] be the resistance offered by the water and air to the displacement of the ship at speed V, and let $\frac{\partial M}{\partial t}$ in [kg/s] be the corresponding fuel consumption time rate, it has been demonstrated that consumption increments associated with the computed increases in resistance, ΔR, and velocity, ΔV, are given by Eqs.(22) and (23), respectively.

$$\Delta\left(\frac{\partial M}{\partial t}\right) = \frac{k(2a_1+3a_2V\ \ )V}{a_1+2a_2V}\Delta R \tag{22}$$

$$\Delta\left(\frac{\partial M}{\partial t}\right) = k(2a_1 + 3a_2V^2\ \ )\Delta V \tag{23}$$

Eqs. (22) and (23) will be used to estimate the consumption increments, $\Delta(\frac{\partial M}{\partial t})$, associated with the computed increases in resistance, ΔR, and velocity, ΔV, resulting from the effects of the above mentioned environmental conditions. It should be stressed that linearity in the dynamic behaviour of the system in the neighbourhood of V_0 has been assumed, so that the present formulation will only be valid in a context where the effects of wave wind and currents are sufficiently moderate.

The trajectory from point *A* to *B* is planned to be carried out at a ship's advance base speed V_0 to reach the destination within the estimated or desired time. The estimation of the effects of wind, waves, and ocean currents on speed and fuel consumption involved the formulation of some simplifications, namely:

1. In considering the effects of environmental conditions, the principle of superposition will be applied, i.e., implicitly linearity in the dynamic behaviour of the system in the neighbourhood of V_0 and $(\frac{\partial M}{\partial t})_0$ is assumed. Hence, the variation (either increase or decrease) in speed or fuel consumption stemming from environmental conditions will simply be added to the base velocity V_0 and the base fuel consumption $(\frac{\partial M}{\partial t})_0$;
2. In the calculation of wind, drift, waves and current effects it has been assumed that vessel's heading is equal to her course;
3. Vessel drifting associated with wind loads has been neglected, and the wind loads will only impact the increased or decreased resistance to the ship's advance. Consequently, leeway corresponds only to either increments or decrements to the base consumption $(\frac{\partial M}{\partial t})_0$. The increased or decreased resistance to ship's advance is computed using the formalism condensed in Eq.(5);
4. The increases or decreases in speed along the course caused by ocean currents at the free-surface will simply be added to the ship's advance base velocity V_0. The leeway produced by ocean currents will be considered only as an increase in the base consumption $(\frac{\partial M}{\partial t})_0$, i.e., the corresponding increase in the velocity to maintain the ship's advance base velocity V_0 along the defined course over ground;
5. Waves will be characterised by significant wave heigh, H_S, spectral peak period T_P, and a relative incoming incidence angle β, which will induce two additional ship's resistance components: added resistance or wave induced roll motion. These two ship's resistance components will be varying from a maximum to zero depending on their corresponding spreading functions, which are defined by an incidence angle, using the formalisms exposed in Eqs.(9) and (12).

Before numerical simulation results of the global fitness *F* being presented, it should be referred that implementation of the heuristic to minimize Eq.(18) comprises the following main steps:

1. Compute $C_{dir} = C(L_{dir})$ and $T_{dir} = T(L_{dir})$ where L_{dir} stands for the direct rhumb line route from *A* to *B*;
2. Generate a set *U*, of 2*M* different routes L^j from *A* to *B*, each one defined by sequences of *N*-1 yaw points in randomly distributed spatial positions;
3. Compute de fuel consumption $C(L^j)$, the travel time $T(L^j)$, and the fitness $F(L^j)$, on each one of the routes in *U* and retain an ordered subset $V \subset U$ of the routes with the lower global fitness *F*. Typically $\#V = \frac{\#U}{2}$. Note that routes L^j, in V=$\{L^1, L^2, \ldots, L^j, \ldots, L^k, \ldots, L^M\}$, must be ordered such that $F(L^j), < F(L^k) \Rightarrow j < k$;
4. Construct a new set *U* concatenating a new set of *V* with previous set of best performers *V* and apply a randomly uniformly distributed 2-D spatial perturbation of maximum semi-amplitude ε

to the positions of each one of the yaw points that define each of the 2*M*-1 last routes in *U*. The first and better route in the current epoch remain undisturbed and survive to integrate the set *U* in the next epoch without any perturbation imposed in order to prevent that a top performer candidate prematurely detected could be discarded;

5. Repeat steps (iii)-(iv) using in each repetition a smaller semi-amplitude ε of the maximum spatial perturbation;
6. Stop when the best route in *V* route fails to show significant improvements or, after *P* repetitions.

The successive repetition of steps (iii)-(v) allows us to define a sequence *U*(i), *I* =1,…,*P* of route sets and a related sequence ε=ε(*i*), *i* = 1,…,*n*,…,*P* of maximum semi-amplitude spatial perturbation of the yaw points, where each *i* define an "epoch". Moreover, ε=ε(*i*), *i* =1,…,*n*,…, must be a slowly decreasing function in order to ensure an adequate survey of the search space. In particular, the slow decreasing negative exponential function given by the Eqn. (19) can be used,

$$\varepsilon(i) = de^{-\sigma \times i}, \tag{19}$$

d stands for a characteristic length, related to the distance from *A* to *B*, for instance.

The decreasing coefficient σ is given by:

$$d(i) = \frac{ln\left(\frac{d}{\delta}\right)}{P}, \tag{20}$$

δ is small residual distance and *P* the global number of epochs. For *i* = *P*, we obtain:

$$\varepsilon(P) = \delta. \tag{21}$$

The parameters δ and *P* must be selected carefully. The goal is to ensure that the decrease of ε is slow enough and additionally that the spatial perturbations of semi-amplitude δ in the yaw point positions are irrelevant by epoch *P*.

4.1. Calm water power estimations

As it can be seen in *Ribeiro e Silva and Bento Moreira (2024a)*, the power curve in [kW] versus speed in [m/s] obtained from CFD simulations, can be modelled by means of a 3rd order polynomial fit of the type $f(x) = a_1 x^2 + a_2 x^3$, where the coefficients of this polynomial obtained by means of linear regression reads $a_1 = -0.275$, $a_2 = 6.510$.

4.2. Wind load estimations

Firstly, estimation of forces (and moments) caused by wind resistance have been conducted for the containership facing head and beam winds. Next, as shown in Fig.6, trigonometric relations given by Eq.(3) can be used to determine the wind loads coefficients at different incoming angles of direction of the wind relatively to the ship's heading.

Instead of using Ishwerwood, Blendermann or Arianne3D's semi-empirical formulations based simply on lateral and frontal projected areas, a more accurate assessment of the impact of wind on this containership could have been conducted by means of CFD. In that case, CFD presents many advantages, of which the most obvious is adequacy for flow visualization and for design optimisation of all surfaces of the ship above the sea-surface as well as cargo, Fig.7. However, proper selection of the choice of the CFD method as well as a grid convergence study should be adopted in the first place to prevent results becoming affected by the mesh and input data selection.

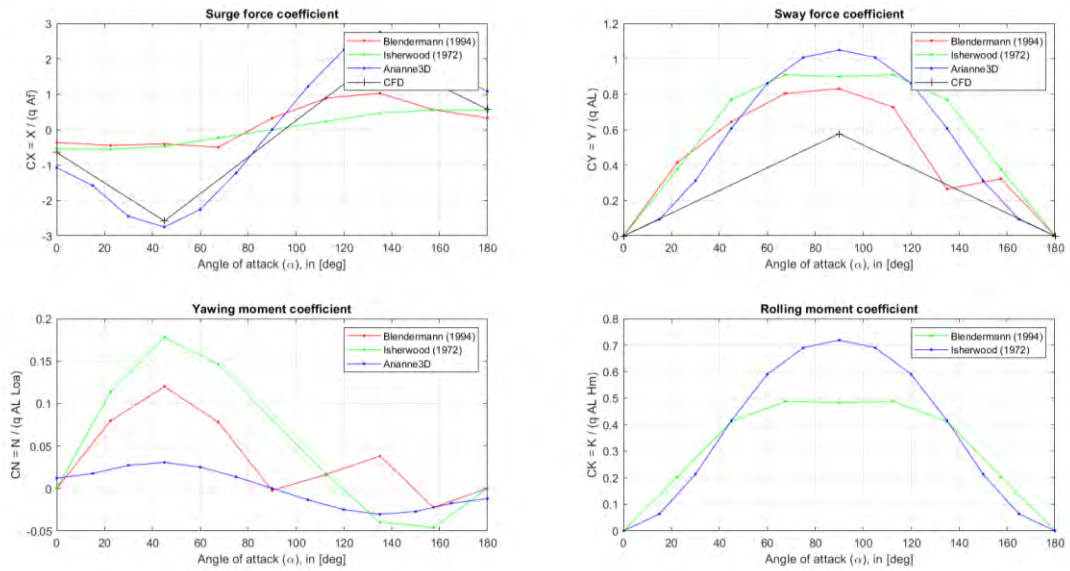


Fig.3: Wind load coefficients versus angle of attack in surge and sway of the containership

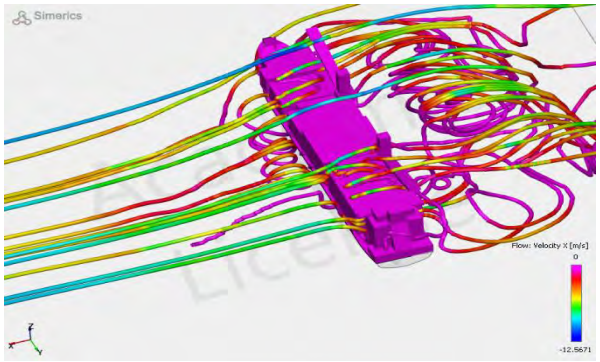


Fig.4: Wind streamlines along the superstructure of the containership

4.3. Trim optimisation simulations using CFD

Use of CFD simulations have been conducted with the aim of obtaining reliable predictions of calm water resistance and trim optimisation calculations at a given ship loading condition. Table I along with Figs.5 to 10 show the power predictions for different trim angles and assuming either constant of about 12,719 t or small variations in the ship displacement.

Table I: Trim optimisation calculations.

Trim angle	Δ varying			Δ constant		
	PE [kW]	Heave [m]	Pitch	PE [kW]	Heave [m]	Pitch
-3°	2854.4	0.186	-0.093°	2784.7	0.181	-0.089°
-2°	2214.8	0.193	-0.004°	2204.9	0.19	-0.006°
-1°	1922.6	0.196	-0.059°	1894.2	0.196	-0.061°
0.0°	1727.8	0.196	-0.095°	1727.8	0.196	-0.095°
+1°	1679.9	0.197	0.114°	1670.8	0.198	0.113°
+2°	1646.5	0.207	0.121°	1644.1	0.207	0.124°
+3°	1793.7	0.221	0.095°	1755.1	0.221	0.095°

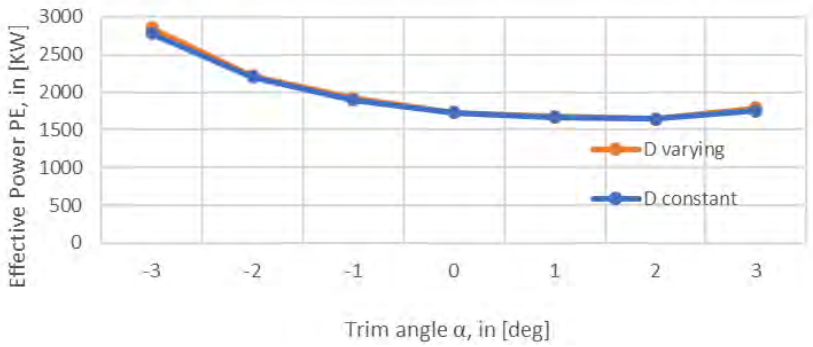


Fig.5: Simerics-MP calculations for power of containership at 14 kn for different trim angles and assuming either constant of about 12,719 t or its small hydrostatic variations due to trim effect

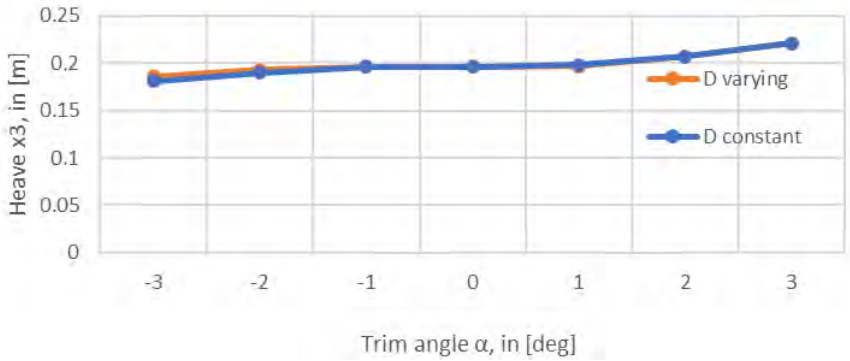


Fig.6: Simerics-MP calculations for heave free-motion at 14 kn for different trim angles

Fig.7: Simerics-MP calculations for pitch free-motion at 14 kn for different trim angles

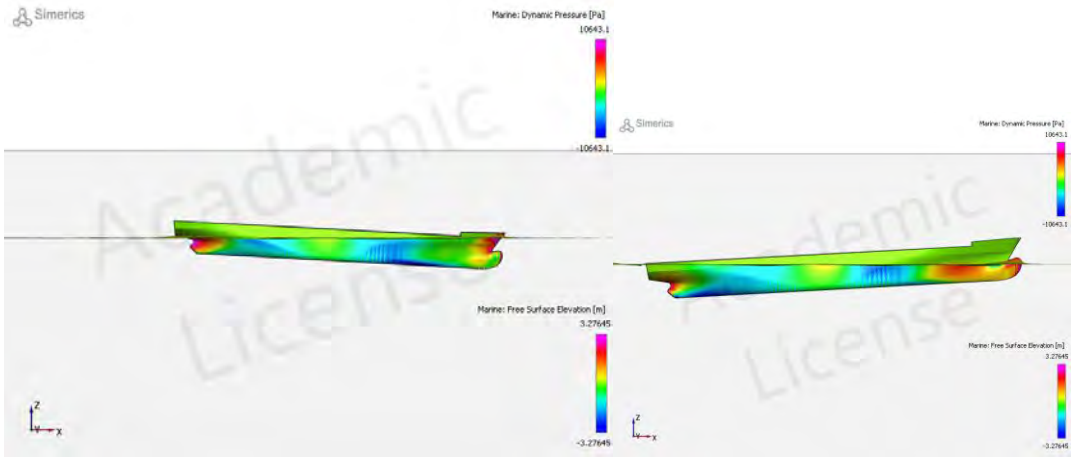


Fig.8: Simerics-MP profile flow visualization of the containership at 14 kn for -3º trim angle (left) and +3º trim angle (right)

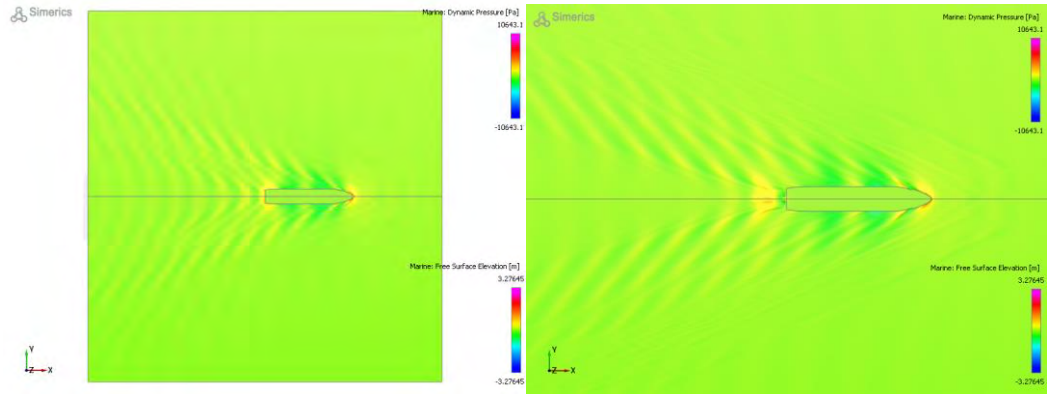


Fig.9: Simerics-MP top-down flow visualization of the containership at 14 kn for -3º trim angle (left) and +3º trim angle (right)

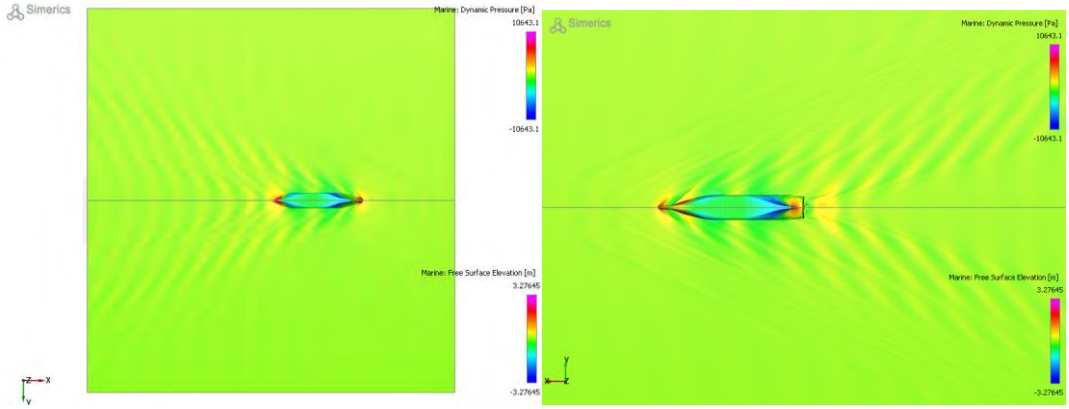


Fig.10: Simerics-MP bottom-up flow visualization of the containership at 14 kn for -3º trim angle (left) and +3º trim angle (right)

Finally, as it can be observed from Table I and Figs.5 to 10 for a given speed of 14 kn, the effective power can be increased 61% or 2% relatively to zero trim condition for trim angles of 3º by the stern or by the bow, respectively. A more useful finding for fuel efficiency is the trim 2º by the bow where a minimum effective power of 1644 kW can be attained, Table I, Fig.5. Hence, it is highly recommended to vessel operators to follow in first place guidance provided by SOOS in order to attain thorough control over LCG position of the vessel that minimizes ship's resistance during operation.

4.4. Operational/numerical simulations using SOOS

In this section, the operating cycle of a containership is defined considering both propulsive load and non-propulsive load. The simulations were based on obtaining optimal route predictions for the 631 nm passage between points *A*(36.95N, 22.5W) and *B*(39.5N, 9.5W). The desired speed V_0 was set to 6 m/s, which corresponds to a desired passage time of 54.14 h (194911 s).

Two sets of simulations were performed: the first from starting point *A* to destination *B*, and the second from *B* to *A*. The results are summarized in Tables II and III, respectively. In each set of simulations, different weighting coefficients were selected, Eqs.(15), to define the relative importance of overall fuel consumption and voyage duration in the objective function *F* (fitness). Thus, the (α, β) pairs used in each of the sets were (100,0), (75,25), (50,50), (25,75), and (0,100), in order to vary the importance of overall fuel consumption and passage time in obtaining the optimal routes.

Figs.11 to 16 show the optimal routes generated by simulations for the following (α, β) pairs from *A* to *B* and from *B* to *A*: (100,0), (50,50) and (0,100). We begin by observing that the optimal routes from *A* to *B* and from *B* to *A* are not identical. This fact is a natural result of the differing effects of environmental conditions relative to the ship's heading.

In both sets of simulations, we observe the following: as the importance of the weighting factor α is reinforced (with a corresponding decrease in the weighting factor β), the overall fuel consumption of the optimal routes decreases, while the total passage time increases. For passages in both directions (from *A* to *B* and from *B* to *A*), by appropriately adjusting the α and β parameters, it is possible to find routes that not only reduce overall fuel consumption but also achieve voyage times compatible with the desired total voyage time. If the arrival time at the destination is not a critical factor, it is possible to find routes that take favourable advantage of the environmental conditions, offering savings in overall fuel consumption that can reach up to 50%.

Table II:Route optimisation from *A* to *B*

	Alpha	Beta	Min. fuel consumption	Min travel time sec/hours
Direct route fuel consumption: **61180 kg** Direct route travel time: **192999 s/53.61 h** Desired travel time: **194911 s /54.14 h**	100%	0%	29530 kg	358054/99.46
	75%	25%	30251 kg	342062/95.02
	50%	50%	57653 kg	196033/54.45
	25%	75%	58661 kg	194090/53.91
	0%	100%	61644 kg	192922/53.59

Table III:Route optimisation from *B* to *A*

	Alpha	Beta	Min fuel consumption	Min travel time sec/hours
Direct route fuel consumption: **19860 kg** Direct route travel time: **197022 s/54.73 h** Desired travel time: **194911 s /54.14 h**	100%	0%	17713 kg	217147/60.32
	75%	25%	17773 kg	213104/59.20
	50%	50%	18647 kg	199369/55.38
	25%	75%	19154 kg	196924/54.70
	0%	100%	19491 kg	196470/54.58

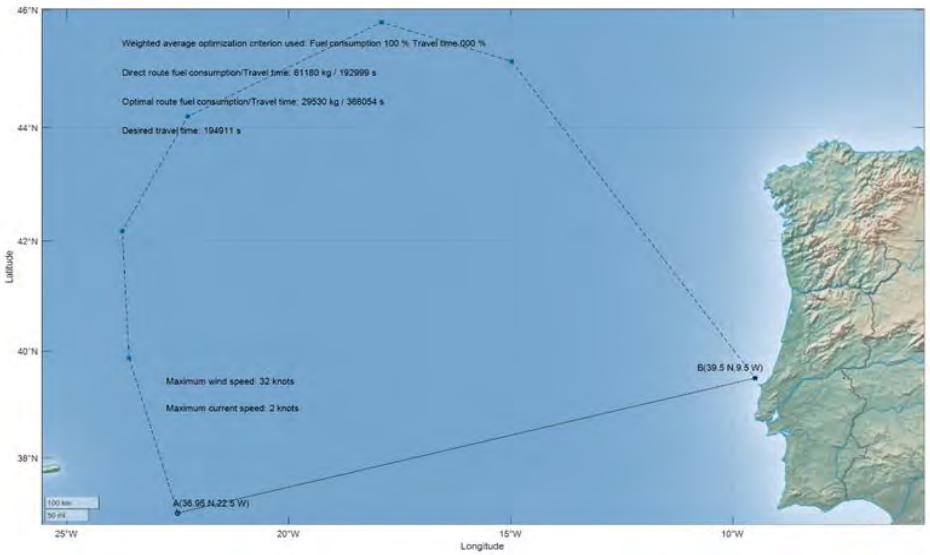


Fig.11: Optimal route in the trajectory AB having a 100% weight factor on fuel consumption reduction and 0% weight factor on travel time duration.

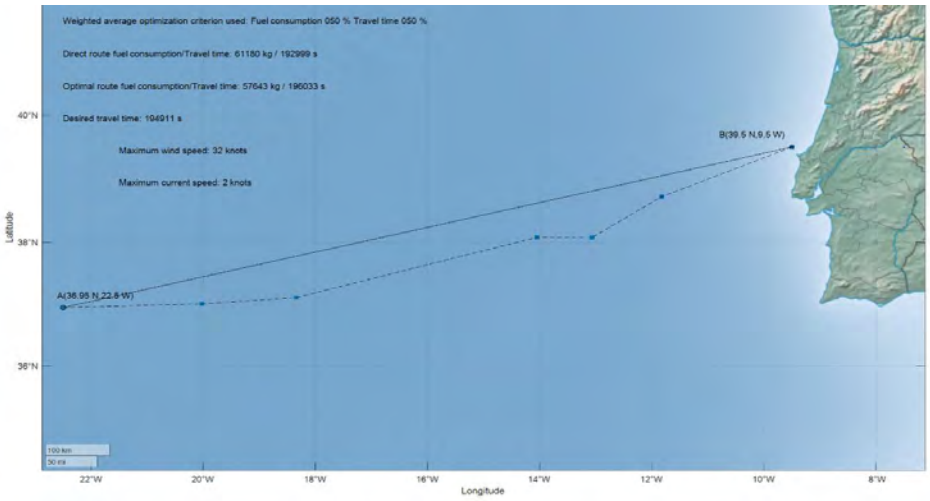


Fig.12: Optimal route in the trajectory AB having a 50% weight factor on fuel consumption reduction and 50% weight factor on travel time duration

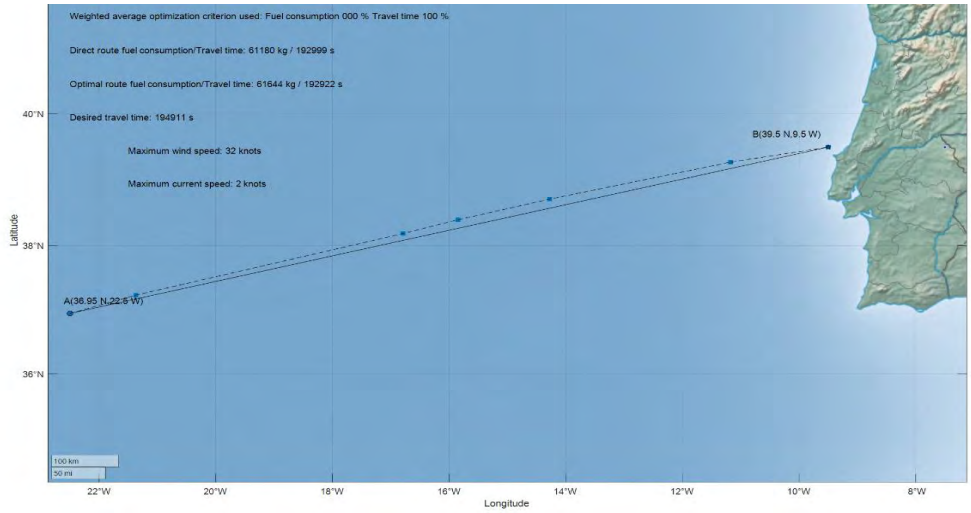


Fig.13: Optimal route in trajectory AB having a 0% weight factor on fuel consumption reduction and 100% weight factor on travel time duration

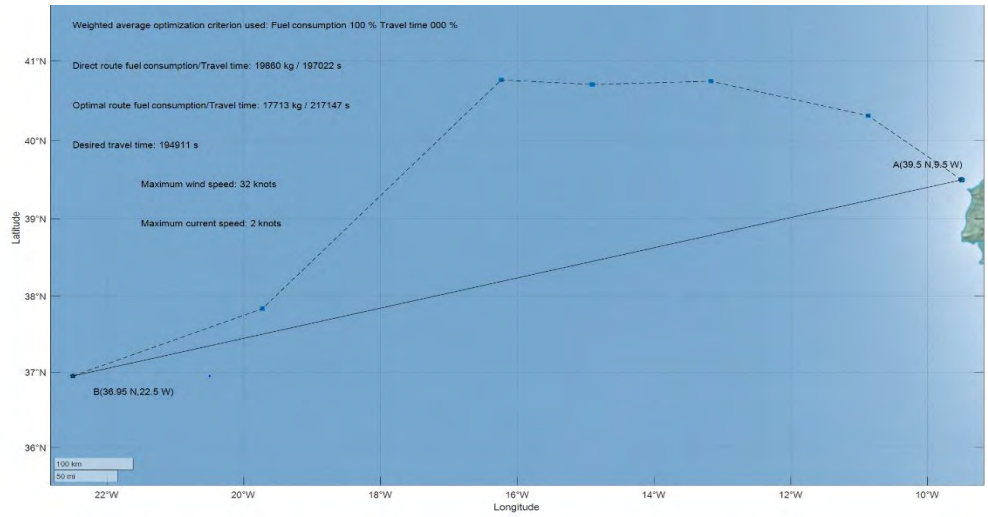


Fig.14: Optimal route in the trajectory BA having a 100% weight factor on fuel consumption reduction and 0% weight factor on travel time duration

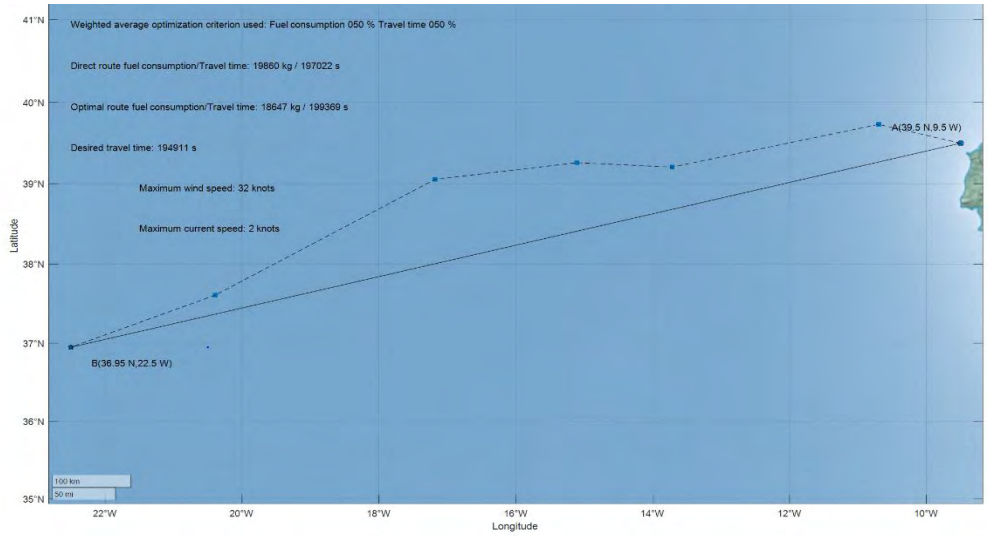


Fig.15: Optimal route in the trajectory BA having a 50% weight factor on fuel consumption reduction and 50% weight factor on travel time duration

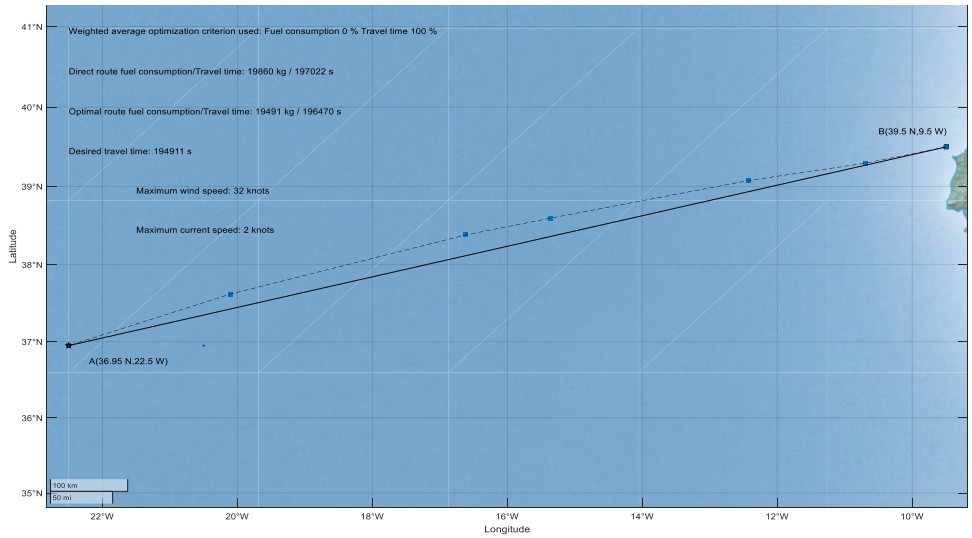


Fig.16: Optimal route in the trajectory BA having a 0% weight factor on fuel consumption reduction and 100% weight factor on travel time duration

The performed simulations are based on the unrealistic assumption that the environmental conditions remain stationary during the passage. This fact suggests numerical simulations to be carried out considering periodic updates of the environmental conditions, using real-time and/or forecast meteorological information.

Considering that effective power can be increased as much as 61% relatively to zero trim for trim angles of 3º by the stern, an additional simulation has been conducted in order to assess the trim effect along the trajectory AB having a 100% weight factor on fuel consumption reduction and 0% weight factor on travel time duration. This numerical simulation revealed that in case an average hydrostatic trim angle of 3º by the stern is kept constant then fuel consumption would be increased by as much as 19% relatively to a level trim condition.

Next the delay of voyages associated with the JiT concept should be further discussed. While slow steaming following SOOS can save fuel consumption of voyages, this would bring additional cost by the increased voyage duration (e.g., additional hiring cost for ship crews and operator). To measure this negative impact of SOOS, the prolongation level of the voyages could be we computed and then subtracted to the fuel and compliance costs savings.

Finally, in the future, the results must be experimentally validated by means of sea-trials, so that simplifications made in the formalism used in here might be appropriately adjusted. Nevertheless, for the time being it is possible to assess the investment of the SOOS + Roll Stabilisation System (RSS) + Weather Routing mid-life refit, as shown in the next section.

4.5. Investment Analysis Of A Mid-Life Refit

In Table IV computation of the Net Present Value (NPV), Internal Rate of Return (IRR) and Payback Period (PP) of our marine transportation energy save project are presented. The retrofitting requires an initial investment of 400 [k€] to install a SOOS + Roll Stabilisation System + Weather Routing instead of running a conventional Very Low Sulphur Fuel Oil (VLSFO) plant whose energy consumptions are 3.5% less (58 [L/mi] of fuel in contrast with 56 [L/mi]). It has been assumed that 48,600 nm will be travelled each year (41.1% of operation rate), the cost of VLSFO is 0.832 [USD/L] with density of 0.875 [kg/L], the annual savings in maintenance associated with a "smoother" propulsion system are 1,000 € and the discount rate for the entire lifecycle of the vessel of 10 years will be 5%.

Table IV: SOOS Investment Analysis

Year	0	1	2	3	4	5	6	7	8	9	10
Investment [k€/yr]	400										
Savings fueland-maint. [k€/yr]		246	246	246	246	246	246	246	246	246	246
Extra cost w/ manning [k€/yr]		-153	-153	-153	-153	-153	-153	-153	-153	-153	-153
Cashflows (Ct) [k€/yr]	-400	93	93	93	93	93	93	93	93	93	93
NPVt [k€/yr]		88.6	84.4	80.3	76.5	72.9	69.4	66.1	62.9	59.9	57.1
NPV[k€]	318.1										
PP [yr]	4.63										
IRR [%]	19.25										

According to Table IV, significant fuel savings allow a payback period of an investment on retrofitting a SOOS aboard a containership of 56 months during the expected lifecycle of 10 years with an IRR of 19%. However, according to recent EU's legislation '*Fit for 55*' 2023 [31] and Emissions Trading Scheme (ETS), unpaid gas emissions savings of around 90 [€/t] of the so-called Carbon Tax should be added to fuel savings in the investment analysis shown in Tab. IV. Note that according to [32] gas emissions can be estimated fairly accurately based on the total amount of fuels combusted and the averaged carbon content of the fuels. Hence, according to our previous estimates, gas emissions savings could represent as much as 26.5 [k€/yr], where, under a complete combustion process an emission factor of 2.7 [kg] of CO_2 per litre of MGO could be defined to perform this greener and more recent investment analysis.

5. Conclusions and Future Work

In this study most recent developments of an optimisation-based approach to reduce fuel consumption along with the passage time of an integrated real-time Ship Operation Optimization System (SOOS) aboard a typical containership are presented. The current study demonstrates that hydrodynamic performance can be substantially improved by means of not only trim optimisation but also wave induced roll stabilisation that allows ships to reduce fuel consumption and simultaneously operate in adverse weather conditions with minimum degradation of their mission effectiveness.

Firstly, from a set of real case stationary environment some preliminary figures on the performance of this newly developed voyage planner based on Vectorised Simulated Annealing (VSA) method was assessed in terms of fuel consumption as a function of speed over ground due to ocean currents and wind loads along with the concept of "just in time" arrival to port.

Finally, these most significant fuel savings in conjunction with the extra cost associated with manning costs allow a payback period of an investment on retrofitting a SOOS aboard a containership of 56 months during the expected lifecycle of 10 years with an IRR of 19%.

In the near future, the performance of the SOOS in terms of fuel consumption reduction will be compared and contrasted against other commercial tools. Contrarily to other decision-support tools systems, SOOS will provide enhanced energy efficiency and roll stabilisation at zero as well as at any advance speed. Therefore, SOOS will be a very attractive option for vessels performing operations that may require a large range of speeds.

Looking ahead, there is significant potential for innovation in this field of providing the maritime transport sector with customised decision-support systems, and further applied RandD in conjunction with anti-rolling U-type tank manufacturers is necessary to further develop this potential. This is reinforced by current trends toward increased automation.

References

BLENDERMANN, W. (1994), Parameter Identification of Wind Loads on Ships, J. Wind Eng. and Industrial Aerodynamics 51, pp.339-351

DNV (2022), *Maritime forecast to 2050 – energy transition outlook 2022*, Technical report, DNV, Hovik, https://www.dnv.com/publications/

GERRITSMA, J.; BEUKELMAN, W. (1979), *Analysis of the resistance increase in waves of a fast cargo ship*, Technical Report 169S, Netherlands Ship Research Centre

GERSHANIK, V.I. (2011), *Weather routing optimisation – challenges and rewards*, J. Marine Eng. and Technology 10(3), pp.29–40

GOULD, R.W.F. (1982), *The estimation of wind loads on ship superstructures*, RINA 8, pp.:1–34

HARVALD, S.A. (1983), *Resistance and Propulsion of Ships*, John Wiley and Sons

HIEMINGA, G.; LUMAN, R. (2023), *Synthetic fuels could be the answer to shipping net-zero goals, but don't count on them yet*, Technical report, https://think.ing.com/

HSVA (2020), *Development of an automated test procedure for efficient determination of roll damping of ships equipped with bilge keels (autoroll)*, Technical Report 1695, HSVA, Hamburg

HITCHCOCK, D.B. (2003), *A history of the metropolis–hastings algorithm*, The American Statistician 57(4), pp.254–257

IEA (2023), *World energy outlook 2023*, Technical report, Int. Energy Assoc., https://iea.blob.core.windows.net/assets/ed1e4c42-5726-4269-b801-97b3d32e117c/

IMO (2023), *The 2023 IMO strategy for the reduction of greenhouse gas emissions from ships*, Res. MEPC.377(80), Int. Mar. Org., London

IPCC (2006), *Guidelines for National Greenhouse Gas Inventories*, Intergovernmental Panel on Climate Change

ISHERWOOD, R.M. (1972), *Wind Resistance of Merchant Ships*, RINA Trans. 115, pp.327–338

LARSSON, A. (2023), *Containers lost at sea 2023 update*, Technical report, https://www.worldshipping.org/statements/containers-lost-at-sea-2023-update

LLOYD, A. (1989), *Seakeeping: Ship Behaviour in Rough Weather*, Ellis Horwood

MAURICIO, F.; MOREIRA, M. (2022), *Optimization of sailboat routes under non-uniform wind velocity fields*, Trends in Maritime Technology and Engineering, pp. 391-396, CRC Press

METROPOLIS, N.; ROSENBLUTH, A.W.; ROSENBLUTH, M.N.; TELLER, A.H.; TELLER, E. (1953), *Equation of state calculations by fast computing machines*, J. Chem. Physics 21(6), pp.1087–1092

NEVES, R.J.J. (1985), *Étude experimentale et modélisation des circulations transitoire et résiduelle dans l'estuaire du Sado*, PhD thesis, Univ. Liége

PRESS, W.H. (2007), *Numerical recipes 3rd edition: The art of scientific computing*, Cambridge University Press

RIBEIRO E SILVA, S. (2008), *Instabilidades no Comportamento Dinamico Nao-Linear de Navios no Mar*, PhD Thesis, Universidade Técnica de Lisboa

RIBEIRO E SILVA, S.; BENTO MOREIRA, M. (2024a), *An optimisation-based approach to reduce fuel consumption and emissions from shipping navigation*, IMDC Conf.

RIBEIRO E SILVA, S.; BENTO MOREIRA, M. (2024b), *An Integrated Real-Time Ship Operation Optimisation System to Reduce Fuel Consumption and Emissions from Shipping Navigation and Port Calls*, ICCAS Conf.

RIBEIRO E SILVA, S.; EÇA, L. (2024), *Solution verification of CFD simulations of a drowning body at sea*, ASME Symp. Verification, Validation, and Uncertainty Quantification (VVUQ), Texas

RIBEIRO E SILVA, S.; VARELA, J.M. (2022), *Ship Gyroscopic Roll Stabilisation*, 41st OMAE, Hamburg

RIBEIRO E SILVA, S.; FONSECA, N.; PASCOAL, R.; GUEDES SOARES, C. (2005), *Motion Predictions and Sea Trials of Roll Stabilised Frigate*, 11th IMAM Conf.

RIBEIRO E SILVA, S.; PASCOAL R.; RODRIGUES, B.; GUEDES SOARES, C. (2006), *Forced rolling trials onboard a PO navy frigate "Vasco da Gama" class*, Marine Technology 43/3, pp.115-125

RIBEIRO E SILVA, S.; GUEDES SOARES, C.; VÁSQUEZ CHILLCCE, G.; MARÓN, A. (2012), The *Stabilizing Effects of U-Tanks as Passive and Controlled Anti-Rolling Devices*, 31st OMAE Conf., Rio de Janeiro

SALVESEN, N., (1978), *Added resistance of ships in waves*, J. Hydronautics 12(1), pp.24–34

SCHEEKLUTH, H.; BERTRAM, V. (1998), *Ship Design for Efficiency and Economy*, Butterworth and Heinmann

STIGTER, C. (1966), *The performance of U-tanks as a Passive Anti-rolling Device*, ISP 13/144, 249

APPENDIX

The vessel studied in here corresponds to a 712 TEU geared containership, whose $Fn = 0.26$. As regards the case-study selection, containerships are the highest emissions producers in the world fleet due to their higher sailing speed that requires larger propulsion engines. Considering the design load condition of 712 TEU, the main characteristics of the vessel are shown in Table A.I.

Table A.I: Main characteristics of the 712 TEU geared containership.

Ship characteristic	Symbol	Value	Units
Length between perpendiculars	Lpp	119.8	[m]
Breadth, maximum	B	20.4	[m]
Draught, mean	T_m	7.75	[m]
Displacement, light condition	Δ	3,936	[t]
Crew	N_{crew}	16	[-]
Gross tonnage	GT	7580	[t]
Cargo capacity	N_{TEU}	712	[t]
Crane elevation capacity (x2)	C_{crane}	40	[t]
Power (effective)	P_E	4665	[kW]
Power at MCR (shaft)	P_S	7200	[kW]
Propeller diameter (CPP)	D_P	4.2	[m]
Cruise speed	U_{cruise}	16	[kn]
Lateral projected area (above MWL)	A_L	1883	[m^2]
Frontal projected area (above MWL)	A_F	444	[m^2]

Fig.A.1 shows a 3D geometric model which has been utilised to perform the CFD resistance and propulsion simulations and the semi-empirical wind loads calculations of the vessel. Considering the origin of the right handed system of coordinates located along the ship keel, in the symmetrical plan, at mean distance between the two perpendiculars, with z-axis pointing upwards and x-axis pointing through the ship bow.

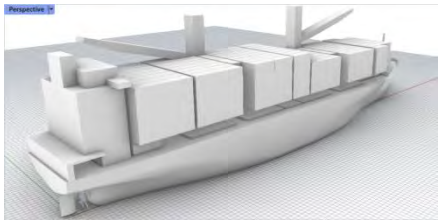


Figure A.1: 3D Geometric Model of the Containership

The main characteristics of the U-tank are shown in Table A.II, and Fig.A.2 corresponds to the geometric model that has been prepared to run the wave interaction analysis software. As it can be observed in Fig.A.2, the U-tank has an adequate fairing to the hull and is located in the exact same longitudinal location as the existing anti-heeling tank in order to minimise the cost of the mid-life refit. In accordance to recommendations of Lloyd [24], the undamped and uncoupled natural period of the fluid inside the tank is 17 [s], which corresponds to a U-tank natural frequency 25% above the ship's natural roll frequency at the design waterline.

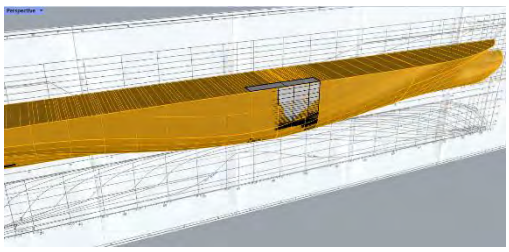


Fig.A.2: 3D Geometric Model of the U-tank between frames #74-#94.

Table A.II: Main characteristics of the U-tank proposed.

U-tank characteristic	*Symbol*	*Value*	*Units*
Duct width	w_d	15.3	[m]
Reservoir width	w_r	2.55	[m]
Tank height	h_t	10.2	[m]
Reservoir height	h_r	2.04	[t]
U tank width	w	16.57	[m]
Duct height	h_d	0.45	[m]
Tank length	l_t	13.15	[m]
Mass of fluid at the tank	m_t	241.5	[t]

From Blueprints to Bytes: OCX is the Future for Model-based Class Approval

Ole Christian Astrup, DNV, Oslo/Norway, ole.christian.astrup@dnv.com
Michael A. Polini, formerly Hexagon and Intergraph, Virginia Beach/USA, mike.polini@gmail.com

Abstract

A model-based class approval and verification scheme for new-build projects requires that the existing 2D drawing-based design documentation be replaced by a 3D digital protocol. The paper explains why the Open Class 3D eXchange (OCX) format is the most viable 3D protocol that can meet the documentation requirements of the classification societies. Today, the classification society's rules state which documents/plans are expected to be submitted by the designer/shipyard for technical review. We review the current documentation requirements in the classification rules and identify and classify the information content in these plans/documents. The content requirements are mapped to the OCX capabilities. Interoperability requirements, such as data integrity and quality, are discussed.

1. Introduction

1.1. History of Standards Development and Adoption in Shipbuilding

In the late 1980s and early 1990s, the U.S. Navy and a consortium of U.S. shipbuilders endeavoured to develop a broad set of STEP-based Application Protocols (AP) for shipbuilding. The Navy Industry Digital Data Exchange Standards Committee (NIDDESC) produced shipbuilding-specific APs founded on ISO 10303, the Standards for the Exchange of Product Data (STEP), covering functions across design, construction, planning and manufacturing (AP 215 – Ship Arrangements, AP 216 – Ship Moulded Forms, AP 217 – Ship Piping, and AP 218 – Ship Structures), *NIST (2001)*. These specifications formed the basis for the later ISO STEP APs for shipbuilding (ISO 10303-215, ISO 10303-216, and ISO 10303-218) embedding lessons learned from the formative years. Although the STEP APs promised a common language, their complexity and the lack of alignment with existing business practices limited their practicality, leaving many protocols underused.

Despite a significant investment of time, money and effort, the STEP initiative never gained broad uptake among CAD vendors or shipyards. Reasons included a weak business case for vendors to develop translators; complex models attempting to capture the entire design lifecycle; fears that easy conversion would encourage yards to switch tools, undermining supplier relationships; over-reliance on shipbuilders with little engagement from vendors or regulators; legacy CAD architectures dating back to the 1970s that could not easily support new protocols; and the concurrent emergence of XML, XSD and 3D API standards that lowered the cost of bespoke, point-to-point translators, lessening the impetus to agree on a single common protocol.

Towards the end of the 20th century, Europe embarked on a collaborative digitalisation programme to standardise data exchange. Backed jointly by the European Union and the European Marine STEP Association (EMSA), it sought to develop, adopt, and prototype standards across the lifecycle - from preliminary design and engineering to procurement and classification approval, producing technical recommendations and scholarly reports to support a model-based shipbuilding process, *EMSA (1999)*.

Several co-funded R&D projects drove the European effort and acted as a testbed for the vision. *NEUTRABAS (1992)* set out to define a neutral product data model for ships and complex systems. Coordinated with IRCN, the French ship research institute, it used the emerging STEP methodology to represent a ship's structure, outfitting systems, and spatial layout in a neutral database. NEUTRABAS delivered concept designs and specifications, and its 1995 report demonstrated that STEP's EXPRESS language could capture the embedded complexity of an entire ship, establishing a reference point for subsequent work. *SEASPRITE (1999)*, an ambitious ESPRIT project driven by end users,

aimed to create an integrated information architecture spanning the entire ship lifecycle – from initial design and simulation through plan approval, construction and operation. Co-ordinated by BMT (British Maritime Technology) with partners including DNV (Norway), LR (UK), Kvaerner, Odense shipyards, MARIN, Kockums and NAPA, it produced key prototypes and contributions to emerging data representation standards, illustrating both the promise and challenges of end-to-end integration.

In the USA, recognition that the shipbuilding industry needed revitalizing led to the National Defense Authorization Act of 1993 and the Clinton administration's plan "Strengthening America's Shipyards*"*. Those documents set explicit goals for establishing a technology base and R&D infrastructure, Shipbuilding Technology and Education (1996), based on an assessment of industry shortcomings. The "Simulation Based Design" (SBD) project under that initiative, redirected focus from downstream "design to manufacturing" towards the upstream "analysis to design" phase, recognising that a fundamentally new paradigm for interface architecture and 3D modelling was required. Funded by ARPA and led by the US-based "National Shipbuilding Research Program" (NSRP), the "MARITECH Advanced Shipbuilding Enterprise" (MARITECH ASE) catalysed collaborative projects between CAD vendors and shipbuilders to explore subsets of a full product model and to lay groundwork for broader digital interoperability, *Gischner et al. (1997,2001), Kloetzli and Billingsley (1990)*.

By the mid-2000s, faced with the cost and complexity of supporting myriad point-to-point links, CAD vendors began to rally for truly "neutral" standards. A schema for a "Neutral XML Interface" was developed and offered to the marine community. Although its uptake remained limited, the exercise matured the discourse and led to a follow-on interface called "Napa Steel", which enabled exchange between the basic design stage and downstream detailed and production stages, which gained some limited support from several shipbuilders. Fig.1 shows a timeline of some major standardisation initiatives.

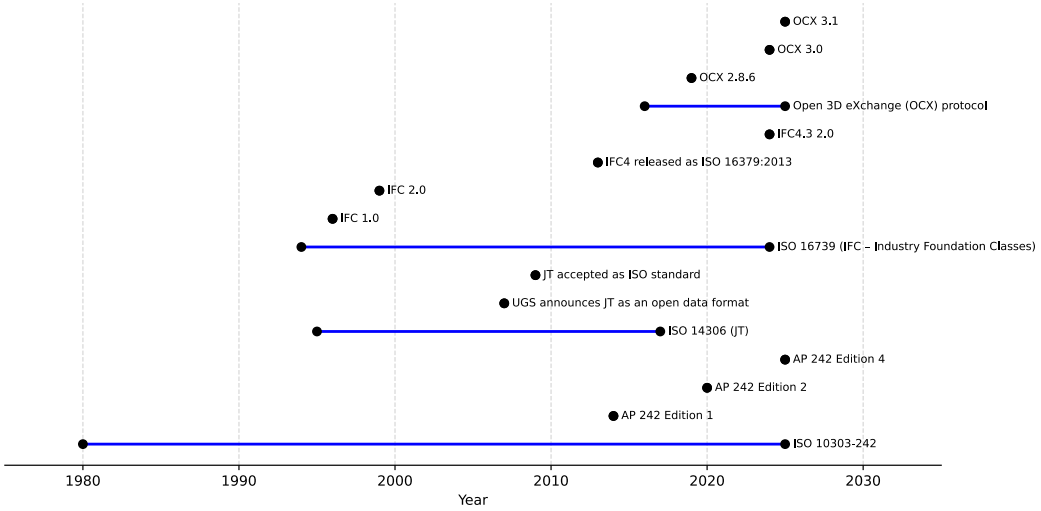


Fig.1: Timeline of major 3D standardisation efforts

Even after decades of digitalisation and standardisation efforts, the shipbuilding industry remains slow to change. While 3D CAD systems are ubiquitous in design offices, 2D drawings continue to dominate many downstream activities. Most yards follow a "3D for design, 2D for production" convention, Fig.2. Designers create detailed 3D models, but 2D drawings are still required for fabrication, assembly, and class approval, meaning that the 3D model is not the single source of truth. Instead, 2D drawings are extracted from the 3D model and manually updated and maintained. This labour-intensive step introduces errors and wastes time. Studies, including work from Chinese shipyards by *Huang et al. (2019)*, confirm that the mixed 3D–2D workflow causes inconsistencies and longer build times. The industry remains far from a true digital thread in which data flows seamlessly from design through construction, operation and end of life. Lessons learned from these initiatives continue to in-

form current proposals and debates about achieving a coherent digital shipyard. The challenge remains to find consensus across diverse stakeholders.

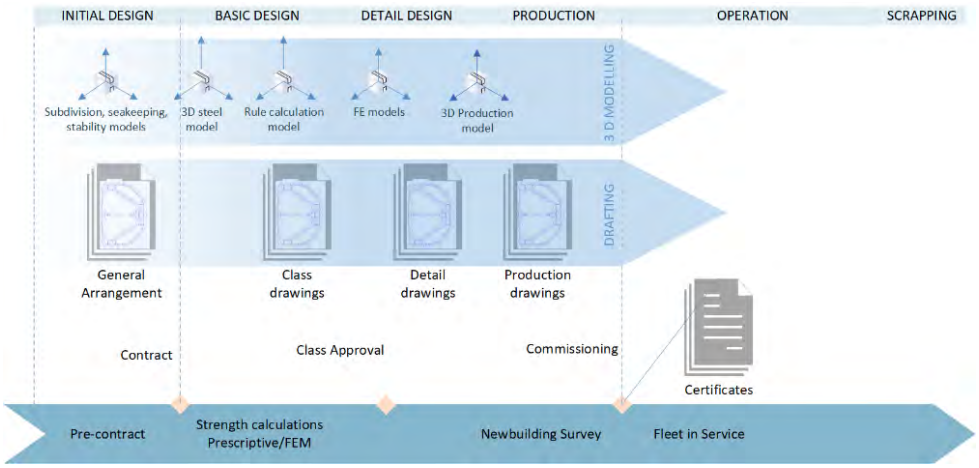


Fig.2: Traditional ship design and construction process and interface to review process by Class

1.2. Major Obstacles to The Missing Digital Thread in Shipbuilding

Past standardisation efforts have highlighted several key barriers:

- Technological and Interoperability Challenges: Ship design and analysis tools often lack interoperability. Stakeholders use different proprietary CAD and analysis software, making data exchange difficult, *Goher et al. (2019)*. Without a widely adopted neutral 3D format —unlike aerospace's STEP AP242 — information is frequently lost or re-entered manually, leading designers to fall back on 2D drawings.
- Process and Legacy Practices: Shipbuilding remains drawing centric. Classification societies, owners, and production teams expect 2D documentation. Moving to digital workflows requires reengineering processes and building trust in 3D models for approval and quality assurance.
- Organisational and Cultural Barriers: The industry's conservative mindset and reliance on 2D hinder adoption. Many professionals lack training in model-based methods, and siloed systems across departments complicate integration.
- Economic and Structural Constraints: Digital transformation demands major investment in tools, infrastructure, and training. With tight margins, especially in commercial shipbuilding, justifying costs is difficult. Unlike aerospace or automotive, shipbuilding has limited influence on CAD/PLM development.
- Legacy Data and Transition Issues: Decades of legacy data in outdated formats pose challenges. Migrating to 3D or maintaining dual systems adds cost and complexity, risking a digital divide between new and legacy projects, *Goher et al. (2019)***.**

What can we learn from previous standardisation efforts? Key takeaways include:

- Clear Business Value
  - Efficiency gains: Less rework and smoother integration between shipyards, owners, and class societies.
  - Cost savings: Standards should reduce lifecycle costs across design, construction, operation, and retrofits.
  - Risk reduction: Improved traceability and compliance with regulatory and class requirements.
- Interoperability & Compatibility
  - Neutral data formats: Must bridge proprietary CAD/PLM/CAE tools.
  - Backward compatibility: Adoption improves when existing workflows remain intact.

- Stakeholder Alignment
  - Shipyards & designers: Need clear benefits such as faster approvals and fewer data conversions.
  - Class societies & regulators: Must formally accept the standard for compliance submissions.
  - Suppliers & owners: Should find value in areas like digital twins, maintenance, and spare parts.
- Practical Implementation Support
  - Reference implementations: Open-source or vendor-supported SDKs, APIs, and libraries to ease integration.
  - Validation tools: Automated conformance checking.
  - Training & documentation: Clear guidance to enable adoption with minimal ramp-up time.
- Governance & Longevity
  - Industry-led initiatives: More likely to establish neutral formats.
  - Open governance: Preferably through ISO or joint industry bodies, not single vendors.
  - Maintenance cycle: Regular updates aligned with industry needs.
  - Clear IP policies: Adoption increases when standards are royalty-free or transparently licensed.
- Proven Use Cases & Pilots
  - Demonstrated benefits: Pilot projects showing measurable savings or smoother approvals.
  - Cross-industry references: Leverage success from aviation, automotive, or offshore sectors.
  - Scalability: Must work for both large shipyards and smaller suppliers.
- Timing & Technology Readiness
  - Cultural readiness: Organisations must be open to shifting from 2D to 3D-centric workflows.
  - Vendor adoption: Major CAD/PLM vendors must support the standard natively.

1.3. The Open Class 3D Exchange – A Fresh Initiative

A major maritime standardisation effort began in 2016, Fig.1, with the Joint Industry Project (JIP) "Approved", led by DNV. Its aim was to establish a standard for digital data exchange in class approval and to develop a neutral 3D format for shipbuilding, *Astrup et al. (2022); Bitomsky et al. (2022)*. The first version of the Open Class 3D Exchange (OCX) schema was released in 2019. Backed by major classification societies and CAD vendors, the initiative led to the formation of the https://3docx.org/en/ consortium in 2021 to advance and promote the format. Key features include:

- A focused scope for practical use
- Broader industry engagement
- Open standard status
- Emphasis on prototyping and test cases

As of this writing, the OCX consortium includes 38 members, spanning classification societies, CAD vendors, shipyards, and design offices.

2. Proposal for a Ship Specific Taxonomy

2.1. What is a taxonomy?

A taxonomy is a structured classification system that organises concepts into hierarchical categories based on shared characteristics, providing a common vocabulary and clear relationships between terms. The main purpose of a taxonomy is:

- Common Language – Provide a shared vocabulary across shipyards, classification societies, owners, designers, and software tools.
- Data Interoperability – Enable seamless exchange of information between CAD/CAE, simulation, classification, maintenance, and digital twin systems.
- Knowledge Structuring – Capture expert knowledge about ship structures, systems, and equipment in a consistent, hierarchical model.
- Traceability & Lifecycle Management – Support linking of components across design, construction, operation, maintenance, and decommissioning phases.
- Decision Support – Provide structured data for analytics, risk assessment, compliance, and optimisation

2.2. Benefits of a Ship-specific Taxonomy

A taxonomy helps manage complexity by enabling consistent data exchange, lifecycle management, and digitalisation — unlocking automation, compliance, and smarter operations.

- Efficiency in Design & Construction
  - Reduces duplication of definitions and miscommunication between stakeholders.
  - Supports automation (e.g., rule checks, material tracking, class approval).
- Regulatory & Standards Alignment
  - Ensures compliance with classification society requirements and other standards.
  - Easier auditing and verification.
- Digital Twin & Simulation Support
  - Makes it possible to map 3D models, loads, and monitoring data consistently to real ship structures.
- Lifecycle Cost Reduction
  - Structured asset information improves spare parts management, retrofits, and recycling.
  - Minimises costly rework due to misaligned definitions.
- Cross-industry Integration
  - Bridges maritime standards with wider engineering/PLM standards (e.g., ISO 15926, STEP, IFC).

2.3. A Ship-specific Taxonomy, Proposal

There are normative vocabularies that define marine and ship-specific concepts. The closest normative references for the shipbuilding industry, relevant to the classification societies, are the ISO 10303 application protocols ISO 10303-215:2004 “Ship arrangement” and ISO 10303-218:2004 “Ship structures”. These STEP application protocols were developed to support the exchange of ship structural models between CAD, analysis, and classification tools. Although the protocols were never widely adopted by the industry, they remain valid normative references covering shipbuilding concepts with clear definitions. Therefore, the protocols provide a sound basis for a ship-specific taxonomy for the hull structure.

The protocols define a broad set of units of functionality (UoF). A UoF represents the highest level and includes one or more sub-levels. We have adopted AP218 as the normative reference, adding only those additional units of functionalities from “AP215 Ship arrangement” that AP218 does not cover. Table I lists the AP protocol UoFs included in the taxonomy. We have selected a subset of the detail levels of UoFs defined in the application protocols to limit the number of nodes in the taxonomy. The detailed UoFs can easily be included as an attribute to the taxonomy node. This approach makes the taxonomy easier to maintain, since attributes and taxonomy nodes can be managed independently.

Table II shows a subset of the ship-specific taxonomy describing the ship cargoes as an example of the taxonomy definition.

Table I: STEP application protocols UoFs included in the taxonomy

Protocol	Unit of Functionality	Incl.	Protocol	Unit of Functionality	Incl.
AP218	class_approvals;	No	**AP215**	arrangement_relationships;	No
	configura-tion_management;	**No**		**cargoes;**	**Yes**
	definitions**;**	No		coatings;	No
	design_loads;	**Yes**		**compart-ment_design_definitions;**	Yes
	external_references;	No		**compartment_properties;**	**Yes**
	hull_class_applicability;	No		compartment_requirements;	No
	hull_cross_sections;	No		configuration_management;	No
	items;	No		damaged_stability;	No
	library_reference;	**Yes**		definitions;	No
	location_concepts;	**Yes**		external_references;	No
	product_structures;	**Yes**		hull_class_applicability;	No
	shapes;	**Yes**		items;	No
	ship_general_characteristics;	**Yes**		loading_conditions;	No
	ship_manufacturing_definitions;	No		location_concepts;	No
	ship_material_properties;	**Yes**		product_structures;	No
	ship_measures;	**Yes**		ship_general_characteristics;	No
	structural_features;	**Yes**		ship_measures;	No
	structural_parts;	**Yes**		**spaces;**	**Yes**
	structural_systems;	**Yes**		surface_representations;	Yes
	welds.	**Yes**		tonnage;	No
				weights.	No

Table II: The taxonomy describing the ship cargoes

taxonomy_id	parent_id	label	Description
ship		Ship	The asset is subject to classification.
cargoes	ship	Cargoes	The cargoes provide the identification of cargoes that can be carried by the ship, applicable properties of those cargoes, and the assignment of those cargoes to compartments in the ship for design or operational analysis.
bulk_cargo	cargoes	Bulk Cargo	A bulk cargo is a type of dry cargo that is solid cargo that is not packed, but is carried loose.
liquid_cargo	cargoes	Liquid cargo	A liquid cargo is a type of Cargo whose natural condition is a non-solid, non-gaseous liquid state.
unit_cargo	cargoes	Unit cargo	The UnitCargo type is intended for spaces carrying a type of dry cargo that is packed or comprises discrete units that can be loaded and stored individually on the ship.

Fig.3 shows the full taxonomy hierarchy as worked out by the authors.

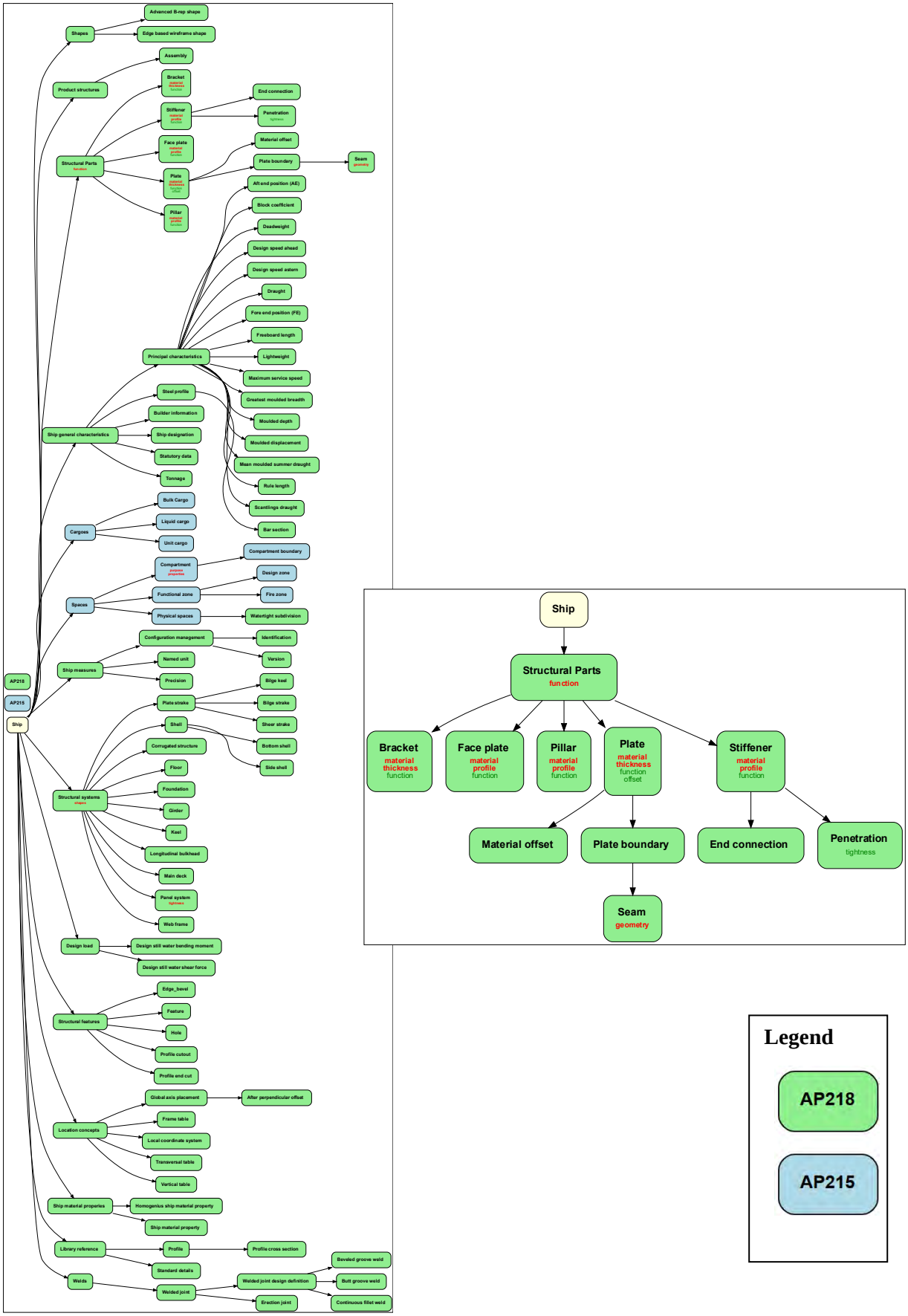


Fig.3: The full ship taxonomy hierarchy (left) and a subgraph (right)

2.4. Mapping the Taxonomy to the OCX Protocol

With the taxonomy defined, it is straightforward to map its items onto the OCX schema by interpreting the taxonomy definitions and finding the corresponding entities in the OCX protocol. The result of this exercise is shown in Fig.4. This exercise demonstrates that the OCX protocol provides a broad coverage of the AP definitions: the OCX accommodates some 85% of the definitions. However, the load and weld definitions are absent because these entities have not been implemented in the OCX protocol. Table III provides a complete list of the missing OCX entities.

Table III: Taxonomy items not covered by the OCX protocol

Dimension	Label	Not mapped ids
cargoes	Cargoes	
design_loads	Design load	design_still_water_bending_moment, design_still_water_shear_force,
ship_general_characteristics	Ship general characteristics	design_speed_astern, lightweight, moulded_displacement, moulded_summer_draught,
structural_features	Structural features	edge_bevel,
structural_systems	Structural systems	keel, plate_strake,
welds	Welds	welded_joint, erection_joint, welded_joint_design_definition, beveled_groove_weld, butt_groove_weld, continuous_fillet_weld,

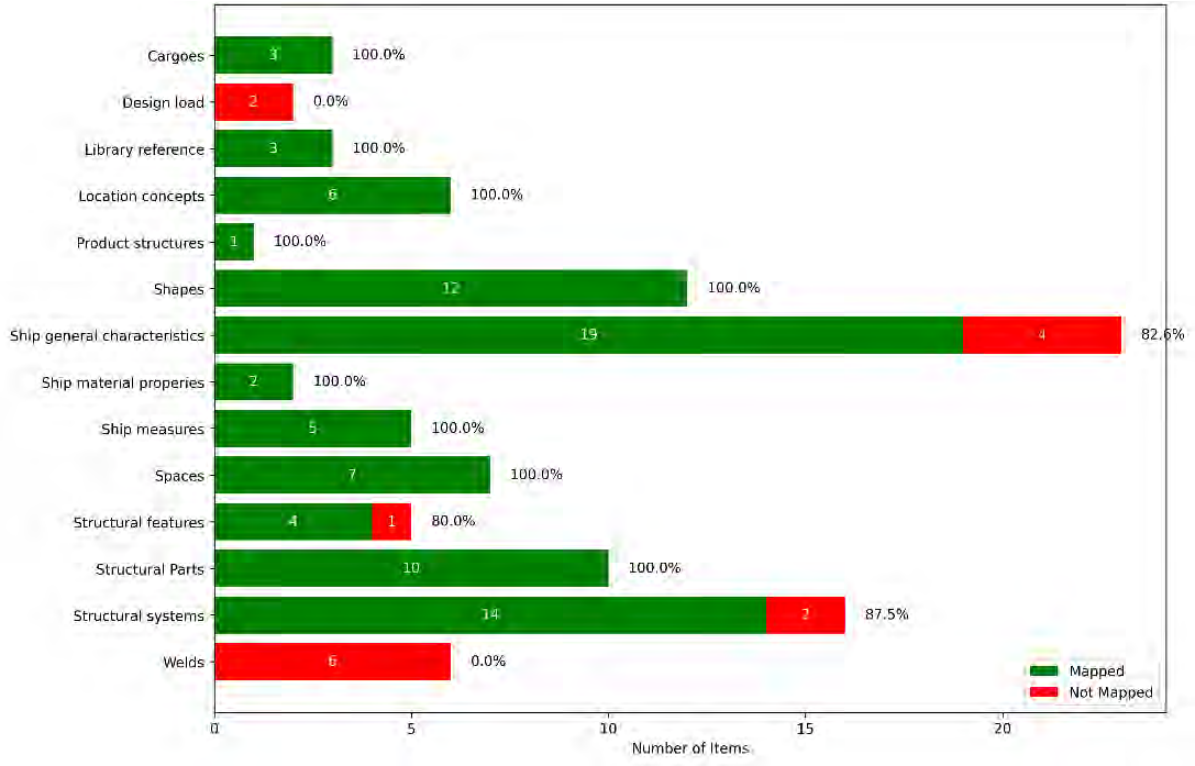


Fig.4: The OCX coverage of the ship-specific taxonomy unit of functionalities (UoFs)

The taxonomy we have employed is far from a complete taxonomy, since it covers only the ship hull structure and derives from a subset of the STEP AP215 and 218 protocols. Its principal purpose is to illustrate the mapping between different sources of information entities pertaining to the ship's hull.

As Fig.4 shows, the OCX protocol covers most of the information entities defined by AP215 and AP218. Given their shared scope, this broad coverage is unsurprising.

Having a normative taxonomy helps to identify information items relevant to the shipbuilding industry. Classification societies could add further value by specifying the requirements for such a taxonomy, identifying which information items they need for approval.

3. Class Approval Use Case

3.1. Model-Based Approval

The intended use case of the OCX protocol is to replace the traditional 2D drawings prepared by yards/designers for design documentation. Until Class approval is given, The Classification Society's key role is to carry out a technical review of the design plans and associated documents for a new vessel, to verify compliance with the applicable Rules and issue the vessel certificates, Figs.5 and 6. In a Model-Based Approval (MBA) scenario, the Classification Society will verify compliance using a digital model instead of traditional paper-based drawings.

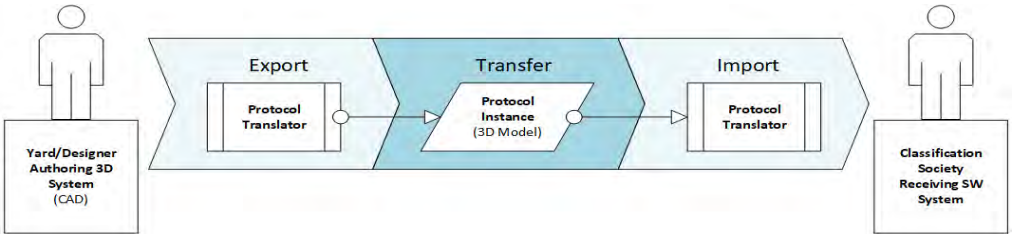


Fig.5: The MBA 3D model exchange scenario

3.2. Classification Review Use Cases

In the context of MBA, *Bitomsky et al. (2022)* describe a generic and simplified design review process consisting of 6 steps, shown in Fig.6.

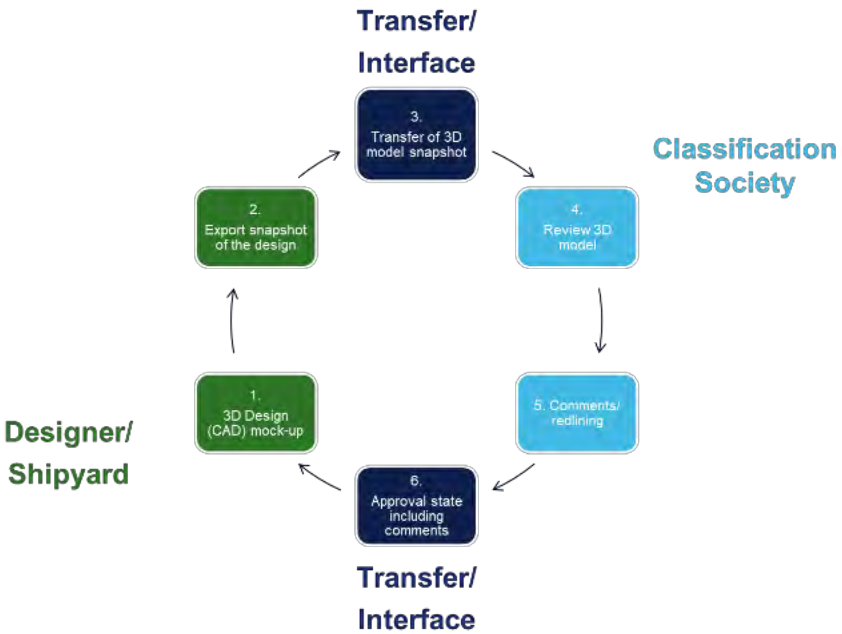


Fig.6: Generic and simplified classification process (MBA context), *Bitomsky et al. (2022)*

The review process also covers visualisation, comment, and red marking. This could be broken down into the following three use cases:

- Use Case 1 – Transfer the 3D model from the Yard/Designer to the Classification society.
- Use Case 2 – Transfer the review results from the Classification society to the Yard/Designer.
- Use Case 3 – Combined Use Cases 1 and 2 as a full round trip.

In the MBA scenario, it is demanded that the 3D model exchange contain all the information needed by the Classification Society to perform a visual verification of the design and necessary calculations or checks to verify compliance. This demand sets the requirements for the engineering data content/MBD. The next section outlines how the engineering data content/MBD is established to support Use Case 1 above.

4. Engineering Data Content for Use Case 1

4.1. Current Class Documentation Requirements

DNV-CG-0550 (2024) lists the formal documentation requirements for the design documents (structural drawings and other documents) to be submitted for verification by the Society. It specifies 41 content requirements for documents related to the hull discipline. Other Classification Societies have similar requirements. 16 of the 41 hull items are drawings; these are listed in Table IV.

Table IV: DNV-CG-0550 list of hull drawings. *DNV-CG-0550 (2024)*

ID	Drawing name
H030	Tank and capacity plan
H040	Structural categorisation plan
H041	Structural inspection plan
H050	Structural drawing
H052	Midship section drawing
H053	Foundation and supporting structure drawing
H060	Shell expansion drawing
H061	Framing plan
H062	Longitudinal section drawing
H070	Standard details
H120	Docking arrangement plan
H133	Erection and inspection plan
H134	Hole and penetration plan
H210	Protected tank location drawing
H220	Cargo safe access plan
H230	Body plan

Table V shows the documentation requirements for a typical “Structural Drawing” (H50) and a specific “Midship section drawing” (H52).

Table V: DNV-CG-0550 document requirements for structural and midship section drawing

Code	Document Name	Content Description
H050	Structural drawing	A drawing showing the geometric dimensions, scantlings and arrangement of a structural object, including: • details of parts and openings • material specifications (see M010 or M030) • standard details (see H070)

		• details of joints, welding procedures, filler metal particulars and specification of heat treatment after welding • inspection category, if not the default category • procedure for stress relieving of cast steel parts.
H052	Midship section drawing	A drawing of the midship transverse section providing information on geometric dimensions, scantlings and material specifications. The following information shall be included on the drawing: • length of ship L • greatest moulded breadth B • moulded depth D • mean moulded summer draught T • block coefficient CB • maximum service speed V • class notations.

The "Structural drawing" (H050) requirements are generic and applicable to all drawings. CG-0550 Section 4 also lists the general requirements for all submitted documentation. A summary of the general information items relevant to the engineering content is listed in Table VI.

Table VI: General document requirements

Code	**Information item**	**Code**	**Information item**
G010	Main class	G020	Asset identification
G010	Vessel type notation	G020	Issuer company name
G010	Class notation	G020	Issue date
G010	Regulatory regulations	G020	Measurement units
G010	Other standards	G020	Revision number

4.2 Drawing Content Classification

When transitioning from 2D to 3D environments, drawing elements must be filtered or adapted. These elements fall into three categories proposed by *Quintana et al. (2010)*:

1. Core Elements – Essential product definition data such as geometry, dimensions, tolerances, symbols, and general notes. These are typically conveyed through orthographic or axonometric views.
2. Peripheral Elements – Supplementary information like class notations, regulatory references, and ship particulars. These apply to the entire product or can be linked to specific geometries.
3. Management Elements – Used for validation, release, change control, certification, and storage. Examples include title blocks, revision history, margins, and versioning data.

4.3. Transition to Model-Based Definition (MBD)

The following drawing elements must be included in a model-based definition or 3D model:

- Core elements must be fully transferred to the MBD dataset.
- Essential management elements should also be included, such as:
  - Identification data: company name and address, dataset timestamp, title and number, builder number, originator's name and date etc.
  - Versioning data: application details, approval records, dataset ID, design activity transfer, and revision history.
  - Required Peripheral elements are included as general attributes or external references.

- Obsolete management elements, such as layout-specific drawing features, are not needed in MBD.
- Adapted elements, e.g., 2D drawing zones (frame table numbering, stiffener Y- and Z-spacings) used for collaboration should be replaced with equivalent 3D zoning annotations to support teamwork in a model-based environment.

Fig.7 depicts an example of a midsection drawing displaying scantlings, measurements, details, zone information, managerial elements (title block, drawing number and revision) and zoning (frame number) with illustration of the four categories.

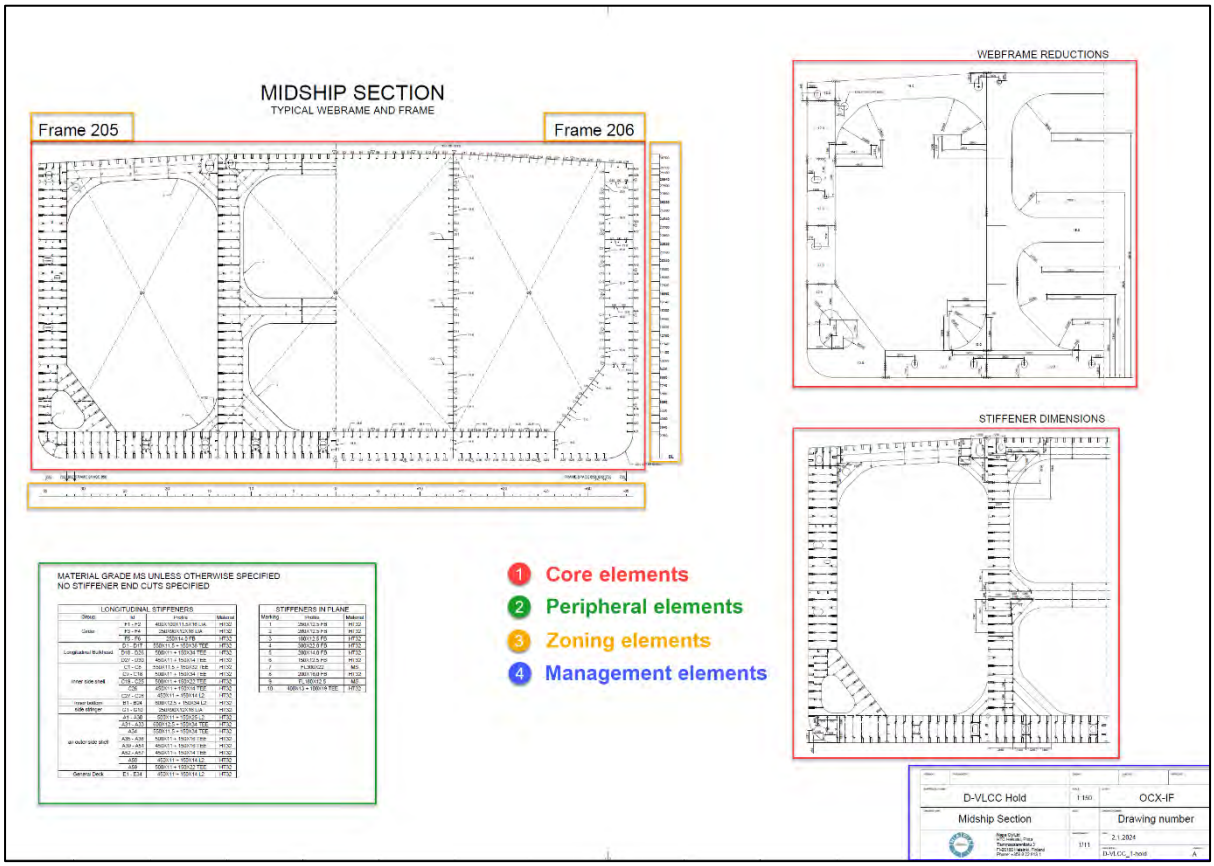


Fig.7: Example of a midship section drawing (H052), source: NAPA

We have used CG-0550 to identify the engineering data content requirements following the principles as illustrated above. The scope is limited to the hull discipline (H) documentation requirements for drawings (plans), excluding DNV Class Notations and service restrictions. The outcome of this analysis identified 73 unique information elements and is shown in Fig.8.

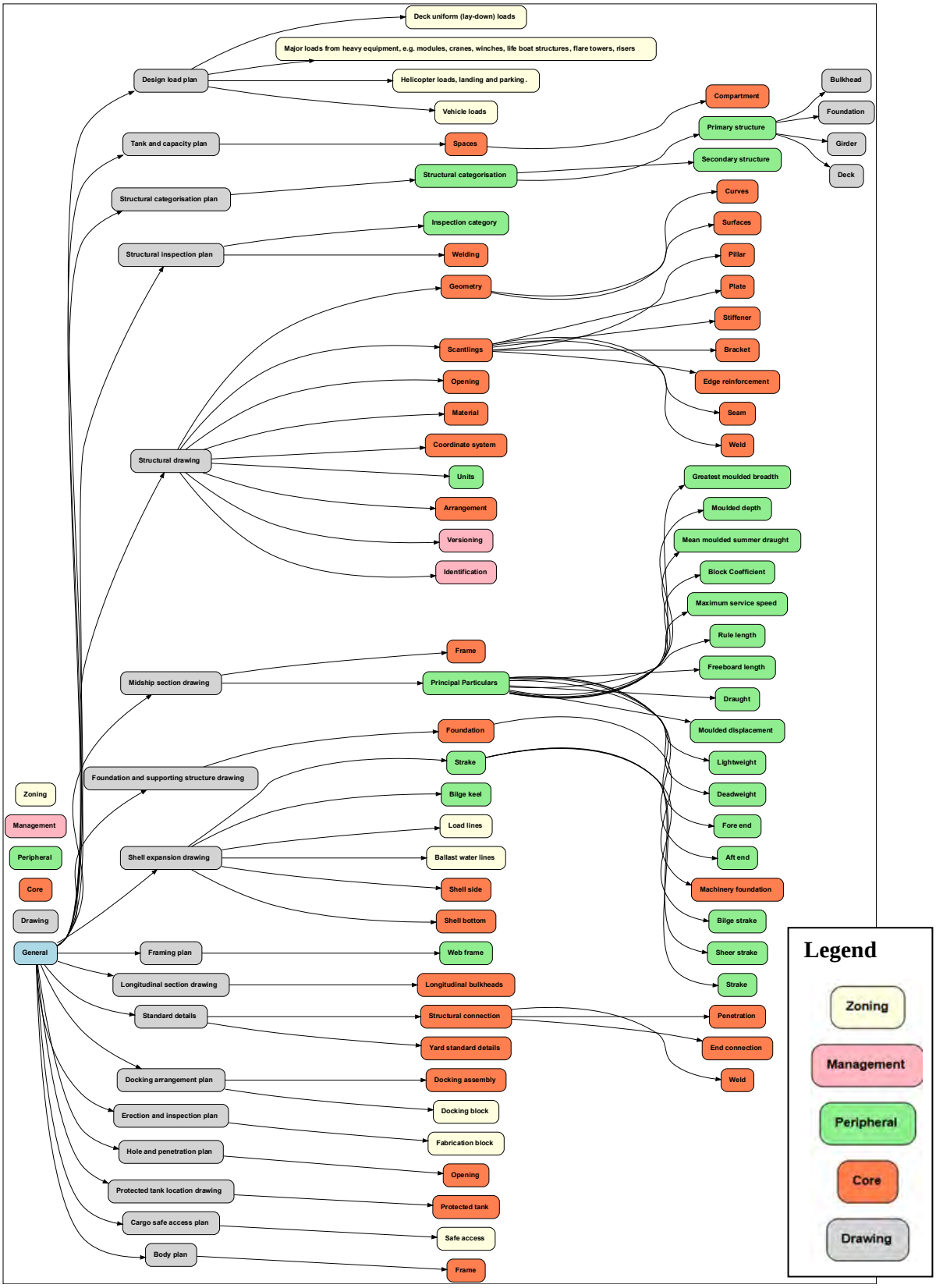


Fig.8: Hull structures drawing content identification and classification (requirements in CG-0550 to other documents than drawings are excluded)

4.3. Mapping of the Documentation Requirements

We mapped all requirement elements shown in Fig.7 to the ship taxonomy described in Section 2. All elements are fully mapped as shown in the chart depicted in Fig.9.

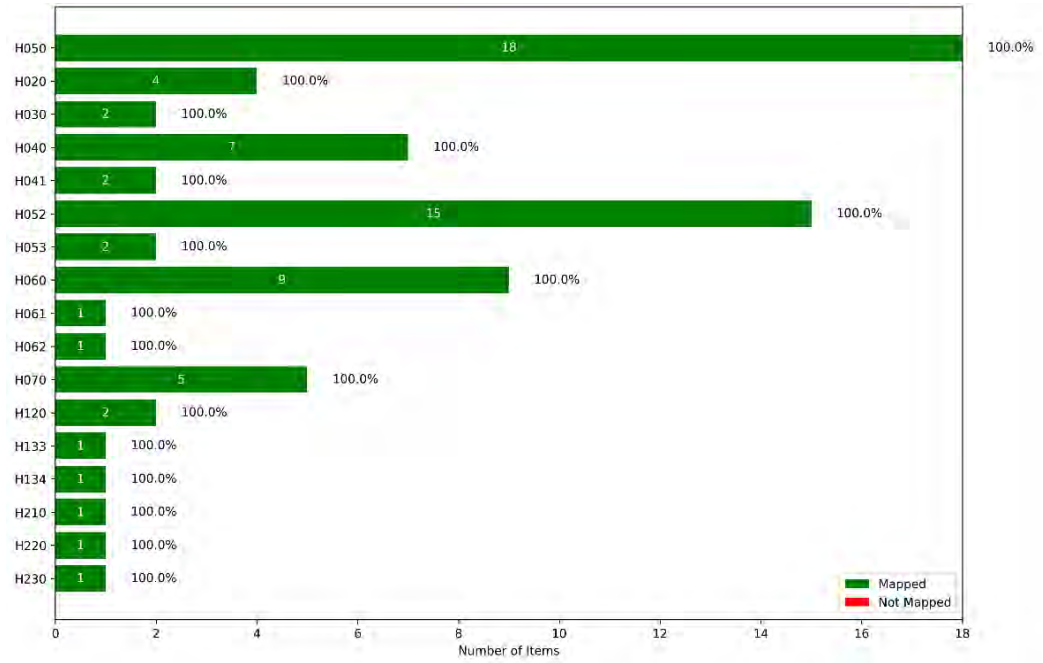


Fig.9: The DNV documentation requirements mapped to the ship taxonomy

4.4. Mapping the Documentation Requirements to the OCX Schema

With a mapping of the documentation requirements to the taxonomy and the mapping of the taxonomy to the OCX protocol, we can now map the documentation requirements to the OCX protocol. The purpose is to identify which of the traditional 2D drawings can be replaced by a 3D OCX model. Fig.10 shows the end-to-end mapping from documentation requirements to the OCX protocol.

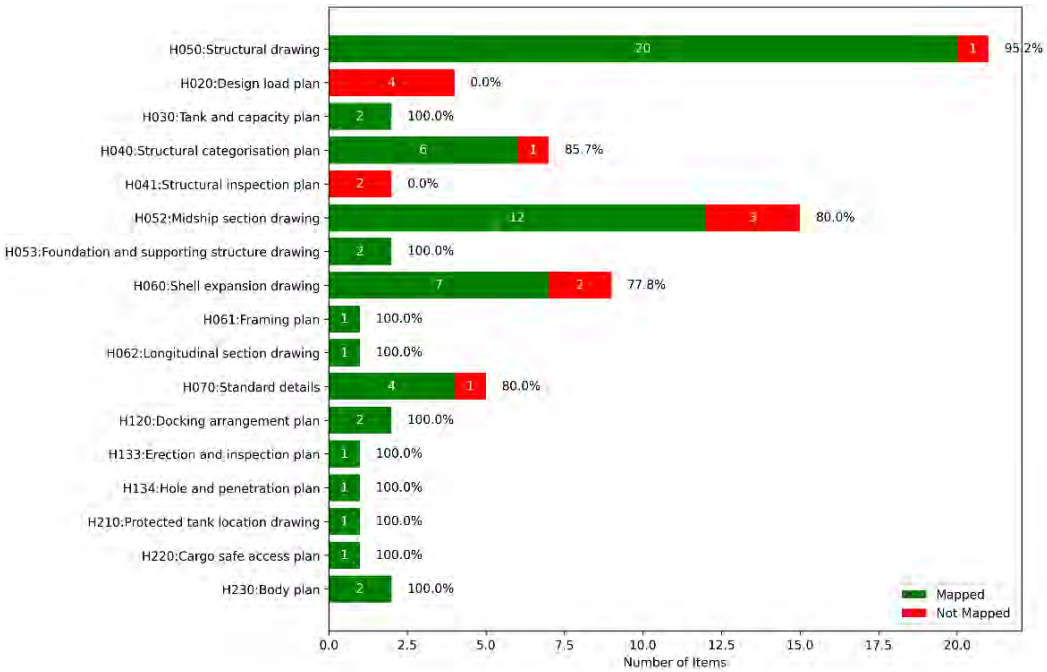


Fig.10: The OCX protocol coverage of the DNV documentation requirements

The OCX protocol covers 85% of the drawing requirements. This directly reflects the OCX protocol coverage of the taxonomy seen in Section 2.4. The difference is that we now directly map the OCX protocol to the Classification Society's requirements in the form of the documentation requirements from CG-0550.

4.5. Real-life Example: Ro-Pax Full Ship Model

A more interesting use case would be to check if a real 3D model contains the information that is required to include in traditional 2D drawings. This is straightforward with the mappings between the taxonomies we now have in place. To illustrate this, we use a Ro-Pax model from NAPA as the use case. The Ro-Pax model is depicted in Fig.11. Fig.12 shows the computed mapping of the documentation requirements coverage by the ro-pax 3D OCX model.

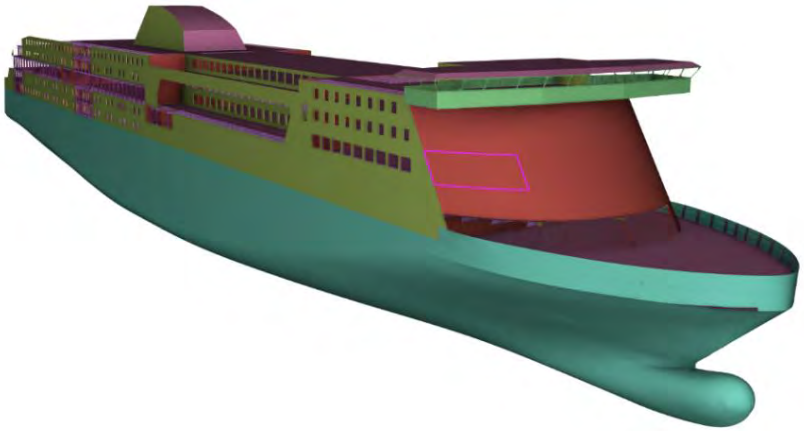


Fig.11: Ro-Pax model, source: NAPA

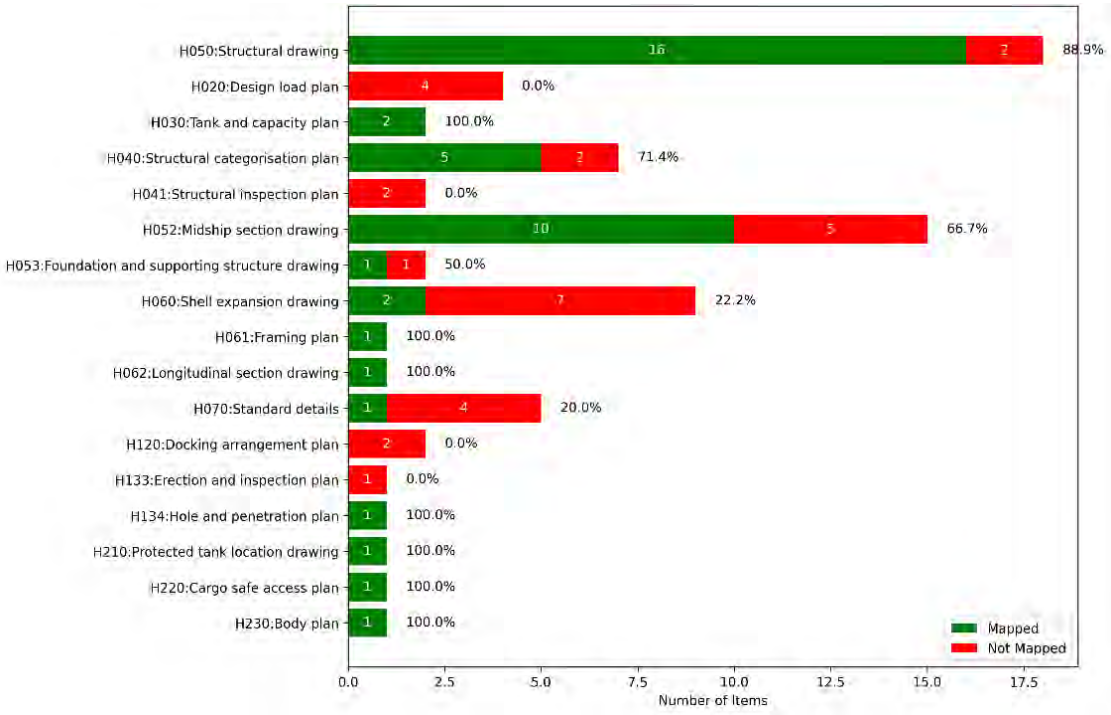


Fig.12: Ro-pax 3D model fulfilment of the DNV documentation requirements

The 3D model covers ~59% of all documentation requirements. We see that the 3D model does not contain any of the information required by the H020:Design load plan, H041:Structural inspection

plan, H120:Docking arrangement plan and H133:Erection and inspection plans. For the remaining drawings, the model partly covers the required information. Table VII lists the missing information elements in the model.

Table VII: Missing required information elements in the ro-pax 3D model

Title	Not mapped ids
H050:Structural drawing	H050.02.04:Bracket (bracket -> Element not found in 3DOCX model), H050.02.07:Weld (welded_joint -> No OCX mapping),
H020:Design load plan	H020.01:Deck uniform (lay-down) loads (design_loads -> No OCX mapping), H020.02:Major loads from heavy equipment, e.g. modules, cranes, winches, life boat structures, flare towers, risers (design_loads -> No OCX mapping), H020.03:Helicopter loads, landing and parking. (design_loads -> No OCX mapping), H020.04:Vehicle loads (design_loads -> No OCX mapping),
H040:Structural categorisation plan	H040.01:Structural categorisation (structural_systems -> No OCX mapping), H040.01.01.03:Foundation (foundation -> Element not found in 3DOCX model),
H041:Structural inspection plan	H041.01:Inspection category (structural_systems -> No OCX mapping), H041.02:Welding (welded_joint -> No OCX mapping),
H052:Midship section drawing	H052.02.03:Mean moulded summer draught (moulded_summer_draught -> No OCX mapping), H052.02.06:Rule length (rule_length -> Element not found in 3DOCX model), H052.02.09:Moulded displacement (moulded_displacement -> No OCX mapping), H052.02.10:Lightweight (lightweight -> No OCX mapping), H052.02.11:Deadweight (deadweight -> Element not found in 3DOCX model),
H053:Foundation and supporting structure drawing	H053.01.01:Machinery foundation (foundation -> Element not found in 3DOCX model),
H060:Shell expansion drawing	H060.01:Strake (plate_strake -> No OCX mapping), H060.01.01:Bilge strake (bilge_strake -> No OCX mapping), H060.01.02:Sheer strake (sheer_strake -> Element not found in 3DOCX model), H060.01.03:Strake (plate_strake -> No OCX mapping), H060.02:Bilge keel (bilge_keel -> not found in 3DOCX model), H060.03:Load lines (functional_zone -> Element not found in 3DOCX model), H060.06:Shell bottom (bottom_shell -> Element not found in 3DOCX model),
H070:Standard details	H070.01:Structural connection (end_connection -> Element not found in 3DOCX model), H070.01.02:End connection (end_connection -> Element not found in 3DOCX model), H070.01.03:Weld (welded_joint -> No OCX mapping), H070.02:Yard standard details (standard_details -> Element not found in 3DOCX model),
H120:Docking arrangement plan	H120.01:Docking assembly (assembly -> Element not found in 3DOCX model), H120.1:Docking block (assembly -> Element not found in 3DOCX model),
H133:Erection and inspection plan	H133.01:Fabrication block (assembly -> Element not found in 3DOCX model),

5. Data Validation and Data Integrity

Demonstrating a broad coverage of the documentation requirements by the Classification Society is not sufficient to replace 2D drawings with a 3D model. A protocol has no value unless interoperability and data quality can be assured and is trusted.

The OCX Consortium has established an Interoperability and Testing Forum (OCX-IF) with the aim to accelerate the delivery of production-level, use-case tested functionality and cross-platform OCX-based workflows to the market. The objectives of the OCX-IF concentrate primarily on testing the interoperability and compliance of Export and Import translators based on the published and draft versions of the OCX Schema, and include documenting and prioritizing use cases, requirements and best practices to ensure completeness and consistency of the OCX Schema and it's implementations, implementing new functionalities based on users' requirements while ensuring these do not adversely affect existing implementations, avoiding roadblocks by establishing agreed-upon approaches, and increasing user confidence in OCX by providing interoperable commercial software products.

The OCX-IF targets two test rounds per year and publishes summary results to the user community. Furthermore, Recommended Practices are developed, and schema issues are reported to the standards development community. Test results are based on comparing "native" (exported) and "target" (imported) statistics, Fig.13 (left). Test results are evaluated for a selected subset of the full data, as defined in the Test Suite. These are known as "statistics". There are two sets - one generated by the exporting system, known as "native" statistics, and another by the importing system, known as "target" statistics. Comparing the values between the "native" and "target" statistics produces the test results. An example is shown in Fig.13 (right).

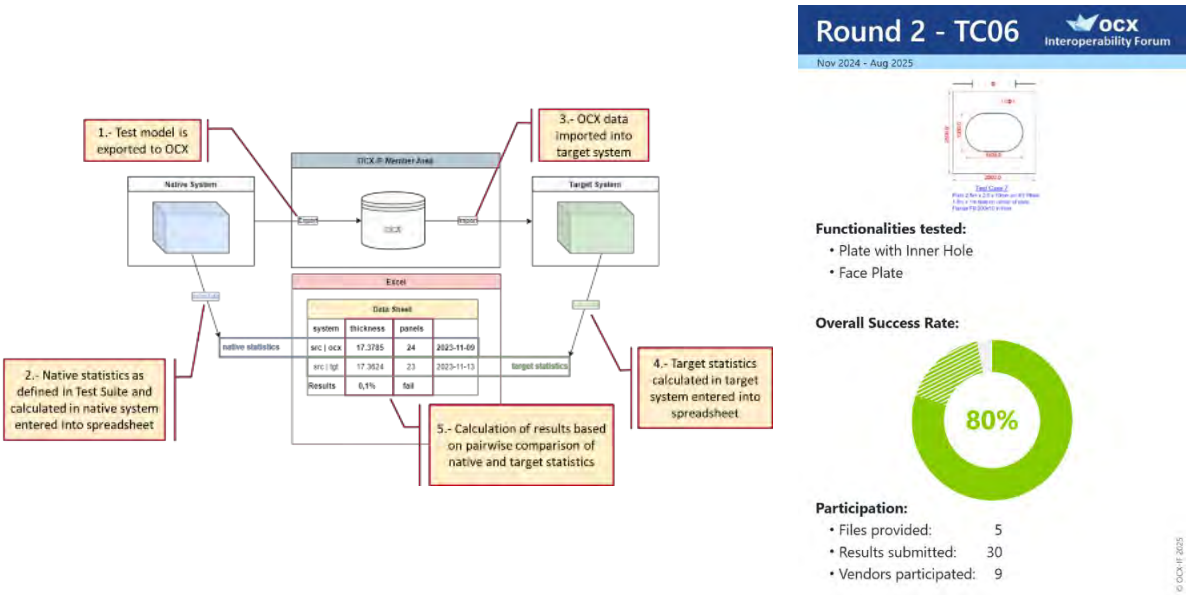


Fig.13: Comparison of native and target statistics to assess interoperability of translators and sample test result

The testing cycle is iterative in nature, and based on an analysis of the test results, corrective actions are identified and assigned to the respective parties. The test results are one of the considerations for schema enhancements and test suite scope for future rounds of testing. The OCX Consortium publishes a full test report at the end of each test round.

6. Conclusions

Shipbuilding is at the cusp of a major transformation. The industry is evolving from a document-centric past into a data-centric future, leveraging model-based definitions as the cornerstone of a more efficient, innovative, and collaborative way of working. While shipbuilding started early on this journey, adoption has been lagging that of the automotive and aerospace industries. Current efforts now are benefiting from their lessons and the maturation of technology.

Taken together, these efforts illustrate a coherent and actionable pathway toward digital transformation in ship classification. The technical groundwork has been laid, the mapping validated, and the infrastructure for continuous improvement is in place. While challenges remain – particularly in achieving full coverage and institutional alignment – the trajectory is clear. The industry stands at the threshold of a more integrated, efficient, and reliable model-based future.

With continued commitment to interoperability standards, cultural change, and stakeholder cooperation, the longstanding vision of having a fully model-driven design and approval process is becoming reality. The title "From Paper to Bytes – OCX is the Future for Model-based Class Approval" aptly captures this trajectory: moving away from paper blueprints to digital bytes, with the OCX standard playing a pivotal role in enabling that future. As this vision materializes, the entire marine industry – from designers and builders to owners and regulators – stands to gain through improved productivity, reduced costs, and better ships delivered in less time.

7. Further Work

We have used a ship-specific taxonomy to verify the foundational elements required for a model-based class approval regime. There is no doubt that the maritime industry will benefit from a normative, ship-specific taxonomy, and the industry should seek to develop this. It must support the needs of classification societies, designers/yards, manufacturers, owners/operators, and other relevant stakeholders. Such a taxonomy can be built and published by targeting one dedicated scope at a time. It should draw definitions and inspiration from existing normative sources, such as the STEP AP protocols, the Skipsteknisk Forskningsinstitutt (SFI) Coding System, the ISO/IEC RDS 81346 standard series, and other relevant ISO standards.

The classification society's feedback to the designer/yard is a missing step in the class approval use case described in Section 3. There is already ongoing work within the industry to develop a solution to bridge this gap, and we will likely see demonstrations of a complete use case soon.

Another key role carried out by classification societies is the construction survey, which verifies that the ship is built according to the approved design. In a model-based class approval regime, where drawings are replaced by 3D models, surveyors at the yard will need to be equipped with new tools.

References

ASTRUP, O.; AAE, O.; KUS, T.; UYANIK, O.; BITERLING, B.; BARS, T.; POLINI, M.; VIJAYA, G.; YU, K.; SEPPÄLÄ, T.; SON, M.J. (2022), *Moving towards model-based approval – the open class 3D exchange (OCX) standard*, Int. J. Maritime Engineering Vol. Part A

BITOMSKY, J.; DANETZKY, A.; ZERBST, C. (2022), *OCX Revisited : The New Data Exchange Standard ?,* COMPIT Conf., Pontignano, pp. 429-436

DNV-CG-0550 (2024), *Maritime services – terms and systematics*, DNV, Hovik
https://standards.dnv.com/explorer/document/6344F50C649D4F49A3D4E5884D99D1F7/9

EMSA (1999), *EMSA To Implement New Industry Standards*, *MarineLink*, 12 November, https://www.marinelink.com/news/implement-standards302723

GISCHNER, B.; BONGIORNI, B.; HOWELL, J.; LOVDAHL, R.; VOGTNER, G.; WILSON, A. (1997), *STEP Implementation for U.S. Shipbuilders - MariSTEP Progress Report*, No. NSRP 0532, SNAME, New Orleans

GISCHNER, B.; KASSEL, B.; LAZO, P.; WOOD, R.; WYMAN, J. (2001), *Evolution of STEP (ESTEP): Exchange of Shipbuilding Product Model Data*, J. Ship Production 17/3, pp.151-160

GOHER, K.; ESSAM, S.; AL-ASHAAB, A. (2019), *Challenges of Model-Based Definition for High-Value Manufacturing*, Advances in Transdisciplinary Eng., IOS Press, doi:10.3233/ATDE190006

HUANG, Y.W.; TANG, Z.Y.; ZHANG, X.H.; ZHANG, H.; LIU, J.F. (2019), *Research on the Three-Dimensional Process Design Method of Shipbuilding Based on MBD Technology*, IOP Conf. Series: Materials Science and Eng. 616/, p. 012029, doi:10.1088/1757-899X/616/1/012029

KLOETZLI, J.; BILLINGSLEY, D. (1990), *NIDDESC: Meeting the Data Exchange Challenge Through a Cooperative Effort*, J. Ship Production 6/2, pp.125–137, doi:10.5957/jsp.1990.6.2.125

NEUTRABAS (1992), *Neutral Product Definition Database for Large Multifunctional Systems*, NEUTRABAS project, https://cordis.europa.eu/project/id/2010

NIST (2001), *Ship STEP on a Page*, National Institute of Standards and Technology

SEASPRITE. (1999), Software architectures for ship product data integration and exchange, SEA-SPRITE project, https://cordis.europa.eu/project/id/22390

Information Architecture in Shipbuilding and Shipping: Ship Lifecycle, Digital Models, Twins, and Fragmented Data Thread

Hideyuki Ando, MTI, Tokyo/Japan, hideyuki_ando@monohakobi.com
Ludmila Seppälä, NAPA, Turku/Finland, ludmila.seppala@napa.fi

Abstract

The maritime industry faces persistent challenges in managing data across the ship lifecycle, from design and construction to operation and decommissioning. Unlike other sectors such as automotive or aerospace, shipbuilding suffers from fragmented data flows and limited reuse of digital assets, hindering the adoption of integrated systems like PLM and Digital Twins. This paper introduces a framework that considers ship lifecycle data from both horizontal (timeline) and vertical (data maturity, dimensionality, and ownership) perspectives. By analyzing lifecycle stages and the associated data structures, ownership, and transitions, the study proposes a structured approach to improve information continuity and digital asset management.

1. Introduction

Shipbuilding and the maritime domain inherited many traditions from the craft stage, when expertise played a key role in the outcomes of the activity. The expertise of naval architects, the layout and location of the building yard, the network and location of subcontractors and equipment suppliers, and the availability of a qualified workforce all significantly impact competitiveness in shipbuilding operations. The profitability of the shipyard's operations and high competition were the main driving forces behind the industry's development for decades. Together with a very low investment in the sector from the states, this created a situation where the shipbuilding methodology remained behind compared to industries such as aviation or automotive. This resulted in significantly less standardized processes, especially in data usage, which made a substantial difference in the digital transformation. The development of IT hardware and software solutions enabled this transformation. While, the work processes and the workforce's ability to adapt, coupled with the fragmented data threads in a long lifecycle, hinder adoption.

Digital transformation in shipbuilding and shipping has the potential to significantly impact the maritime industry. It is expected to bring resource optimizations – including investments, workforce, and work processes - and provide data-driven decisions to support sustainability and carbon neutrality goals. At the heart of this transformation is the efficient and consistent use of information. Currently, the generation and use of data in shipbuilding are purpose-driven for specific parts of the lifecycle and desired outcomes. It is seldom reused in subsequent stages, often based on manual inputs and outputs, and the concept of digital asset management, which involves the systematic organization, storage, and retrieval of digital data, is only beginning to enter the industry.

This paper aims to review the lifecycle stages of ships and the types of data generated at each phase, outline existing ontologies, taxonomies, and standards for maritime data, analyze the role and limitations of digital models and digital twins, and propose a framework that connects lifecycle and data to enable the possibility of digital asset management for the industry. By bridging the gap between traditional shipbuilding practices and modern data methodologies, the proposed framework seeks to enhance lifecycle information continuity and unlock the full potential of digital transformation in the maritime domain.

2. Life cycle timeline for ships and data

The starting point of a lifecycle is to establish the timeline and identify the main phases through which the typical ship goes on its so-called cradle-to-grave journey. The lifespan of vessels in the commercial fleet typically spans 25-30 years, presenting a significant challenge in itself, as

technology and operational modes evolve significantly during such a prolonged lifespan. The design and construction phase for complex modern merchant vessels that utilize alternative fuels and energy-saving devices can take up to 4 years or longer, in the case of cruise vessels. Furthermore, the series production of the same vessel design may continue for nearly a decade and sometimes even over 20 years, starting with the first vessel. Consequently, this business convention also makes it challenging to keep pace with modern technology, particularly in terms of data usage and storage. It presents specific challenges for investment strategy, as typically the investment in developing the concept and design should be allocated before delivery to enable design activities before any tangible delivery to the shipowner.

It is common practice for shipbuilding projects to involve a relatively high number of collaborators, including naval architects, engineers, designers, procurement staff, construction planners, and subcontractors, who often create and utilize their own data and models for specific purposes. Additionally, shipowners and classification societies also add their own information to the data models provided by shipyards. It is coupled with the absence of the main interested party to collect and maintain all related data among key stakeholders; naturally, the shipyard can be such a main contractor party; however, the shipyard's focus often remains on collecting data relevant only to the building and construction process. The lack of methodology and standards for its organization, combined with the absence of digital asset management processes, results in significant unclaimed benefits from digital transformation. The following subsections present views on life cycle stages from both a maritime perspective and a generic PLM and data perspective, highlighting the underlying concepts and similarities in the approach across different industries.

2.1. Life cycle phases in ship design and shipbuilding tradition

At first, we aim to establish a common ground for the ship life cycle. Currently, there are various ways to present this timeline from different stakeholder perspectives. As this process is very complex and multi-stakeholder, it can be divided differently based on the focus of each activity, thereby presenting a particular challenge for unification. Fig. 1 illustrates some of the most common perspectives on the lifecycle from four major stakeholder groups: shipowners, naval architects, shipyards, and a commonly used definition of the life phases.

The high-level perspective of the shipowner would emphasize the main phases, such as requirements specifications, contract, design, construction, and operations. Each of these phases produces a specific output, and there may be differences in the stakeholders involved. In some cases, the shipowner may collaborate with a shipyard that provides a full range of services or even owns such yards. In other cases, there might be many different actors involved, such as naval architecture firms specializing in a specific part of the ship design, design contractors to deliver a design project for particular areas or systems, or parts of the detailed design project, various subcontractors for manufacturing, assembly, and construction, as well as different operators and ship managers.

From a naval architecture perspective, there is a significant emphasis on methodological studies and research, especially in the initial stages of the lifecycle. This focus is driven by the need to effectively manage the complexity of the design process, which is characterized by significant uncertainties and iterations in calculations and simulations. In methodological studies and research, naval architects collect a deeper understanding of the design process, enabling them to make more informed decisions and optimize the design of the vessel. Research publications provide clarification on the ship design process and optimization, *Papanikolaou (2014*), where each activity is represented in a structured manner, including inputs, outputs, and dependencies. The famous design spiral, as described by *Evans (1959),* represents the accumulation of data over the various steps of the design. However, the granularity of this split serves the perspective of naval engineering and therefore doesn't contribute significantly to the overall timeline. A growing body of research addresses the "fuzzy front end" of the lifecycle – where intent and requirements are clarified and converted into design solutions, *Brett et al. (2025)*. This provides the possibility to define the purposes of the designs and align the intent, technology, and economic aspects, often employing methodologies from business research or service

design. It is a relatively novel approach that employs methods from other industries in maritime, and therefore utilizes more modern approaches to work with uncertainty, complexity, and data, resembling those used in software development, such as Agile or DevOps.

The shipyard's perspective focuses on the part of the lifecycle where the shipyard is involved, often leaving out the initial design and operations phases. Another existing perspective is on the management of shipyard activities, *Bruce (2021)*. This process perspective considers not only inputs and outcomes, but also the softer aspects of management and organizational structure involving the stakeholders. It is the most "business-oriented" perspective, and it is often considered an Intellectual Property (IP) of the shipyard, since it can significantly impact the competitiveness of the operations. This perspective can be evaluated relatively separately from the entire lifecycle timeline, as it addresses the operations of the yard, which builds more than one vessel at a time, and therefore is more closely related to the shipyard's efficiency than to the life cycle of each ship built.

Alongside these three main perspectives, a commonly used scale with typical phases is also available. Typically, it incorporates main phases as these are used in the global ship design, building, and shipping operations, and allows for a more detailed breakdown if needed to address a particular part of the lifecycle.

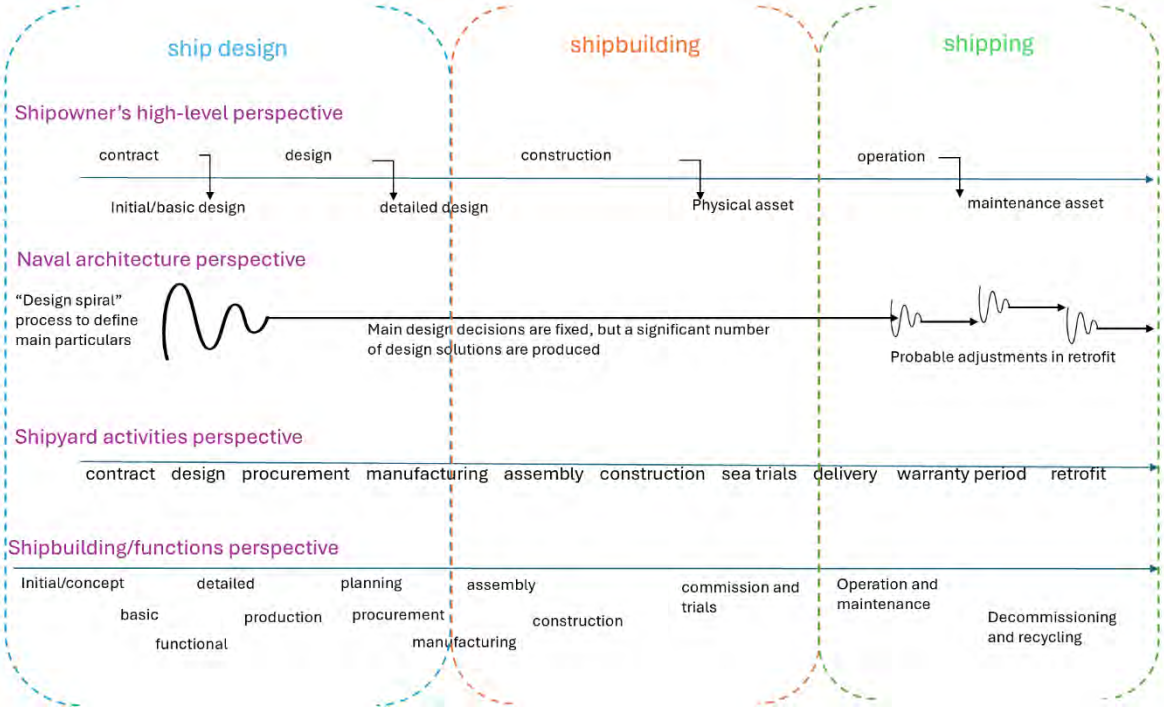


Fig.1: Phases of the life cycle from different perspectives are based on a commonly accepted division between ship design, shipbuilding, and shipping phases in the maritime industry

Navy industry traditionally had longer development timelines and significantly less budget pressure; hence, there are more developed methodologies, Fig.2. It represents similar life cycle stages as those described in the shipbuilding functions perspective above, but also adds a second dimension of maturity, which is linked with milestones of required data maturity for the subsequent stages of contracts and funding. It is a more clearly defined process model of the lifecycle, as the control and auditing of Navy contracts require a comprehensive methodology. The second dimension for the maturity of the data produced provides a possibility to visualize the gaps between the stages. It also shows that the common perception that data slowly flows from one stage to the next is a misconception, as stages often overlap, and input data is incomplete at the start of the next stage.

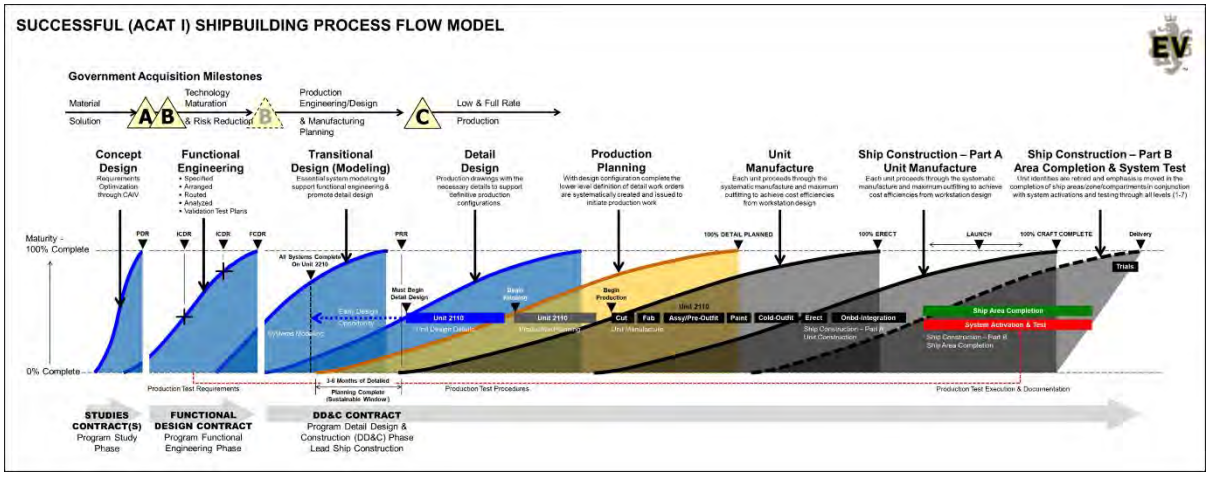


Fig.2: Example of a shipbuilding process flow model, specialized for navy projects, presenting the horizontal timeline and vertical dimension of data maturity, *Hitchcock (2021)*

The perspectives presented above vary in level of detailing and purposes they serve; however, they highlight the main parts of the lifecycle: ship design, shipbuilding, and shipping. Suppose these main steps are taken as the primary stages; in that case, the detailed split and perspective of various stakeholders can be aligned by either splitting into more sub-stages or by adding milestone points, such as ship approval, delivery, or similar. In the following sections, we discuss the lifecycle timeline for other industries and highlight the similarities that can be applied, particularly from the perspectives of data creation and utilization.

2.2. Life cycle phases from a product development perspective

Undeniably, the ship design, shipbuilding, and shipping industry has unique needs and established ways of working. Without losing focus on the specifics of shipbuilding, it is, however, possible to look at it from a more general perspective – product development, where the product is a vessel or a ship. Most often, this comparison is made with the automotive or aircraft industries, as products in these sectors share similarities with shipbuilding, including a complex nature, a lengthy design phase, and an extended lifecycle. Additionally, the construction and software development industries can provide valuable input for the methodology of data use across the lifecycle. The common grounds for such comparison are the complexity of the end products, the multiple interests of stakeholders, the need for highly sophisticated data management processes, and long development timelines.

2.2.1. CAD, PDM, PLM, and eBOM-mBom-sBOM

From a methodology and data perspective, Germany has historically led the way in research in these areas, making significant contributions to concepts such as BOM, PDM, PLM, and later Digital Twin and Industry 4.0. These methodologies emphasized the role of data in the design and manufacturing processes before the significant advancements in software and hardware, and perhaps even served as a driving force behind the need to develop more advanced IT technologies. For shipbuilders, this terminology is only beginning to gain use and acceptance within the industry, as it is often too abstract and primarily addresses the methodology of the process rather than the desired outcomes.

One of the fundamental concepts in the data methodologies of CAD (Computer-Aided Design), PDM (Product Data Management), and PLM (Product Lifecycle Management) is the Bill of Materials (BOM). The simple assumption behind this concept is that every complex product consists of parts, which can be listed as separate entities and classified for functional or manufacturing purposes. For car and aeroplane industries it is typical to use the lifecycle perspective, the main types of BOM – eBOM (Engineering Bill of Material), mBOM (manufacturing Bill of Material), and sBOM (Service Bill of Material) correspond to the same life stages, ie, design, building, and operations, as described before and identified as the main stages in Fig.1. The connections between the BOMs for different

life stages are usually referred to as the digital thread. Fig.3 presents a variety of data structures and corresponding systems in this approach – SysLM (System Lifecycle Management), which considers a product and related processes from a systems-of-systems viewpoint, presenting applicable concepts for understanding the connections between data, methods, and existing IT systems, *Eigner (2021)*. While in the big picture, this assumption can be helpful for shipbuilding, it lacks a concept of topological connections, relations, and dependencies between the parts, which often play a critical role in the data structures.

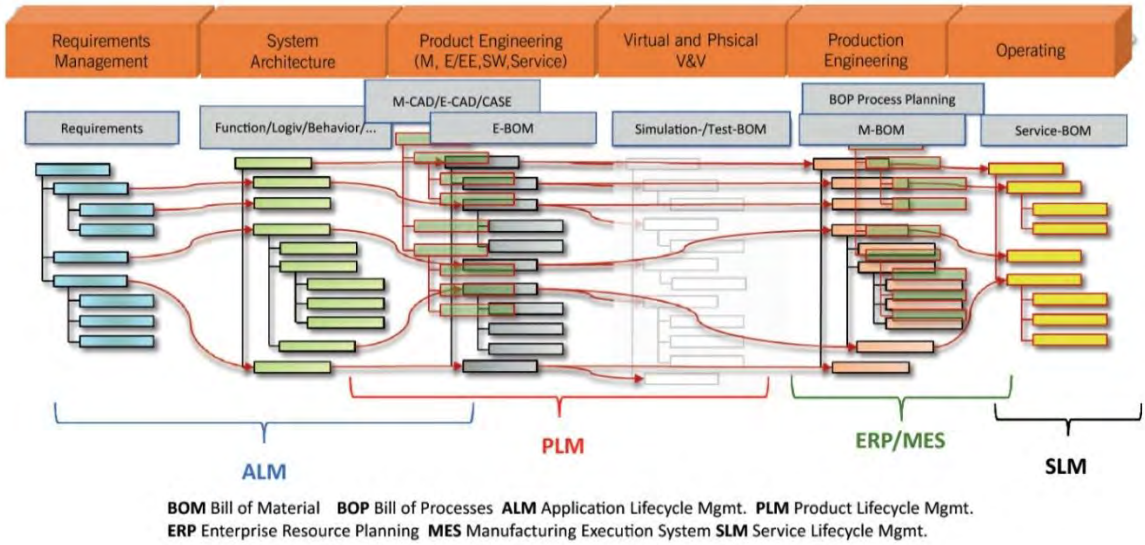


Fig.3: Theoretical PLM data model as a connection of partial models along the product lifecycle, connecting the data with the lifecycle concepts, *Eigner (2021)*

The representation of lifecycle and data in this methodology provides a more in-depth understanding of different datasets and their interactions. It also highlights the core value of PLM systems, not merely as a storage mechanism for various data, but as a facilitator of digital threads that account for specific work processes. In the construction and process plants industry, the primary focus of the lifecycle resembles that of a shipyard, often likened to one, with similar activities to those of an EPC (Engineering, Procurement, and Construction) contractor. Existing frameworks include BIM (Building Information Modeling) and IFC (Industry Foundation Classes), as well as several ISO standards for data exchange and interoperability. Often, the lifecycle itself is standardized and considered more from an LCA (Life Cycle Assessment) perspective in the context of material depreciation and environmental sustainability.

This discussion helps to establish the horizontal direction of the proposed framework. The main phases proposed correspond to ship design, building, and operations, and this alignment is well-suited to the data methodologies, leaving room for more granularity if needed to distinguish separate processes or stakeholders. The following section provides a detailed discussion of the data generated and used at each stage, along with an explanation of how this data is structured and linked.

3. Data ontology, taxonomy, and IP

Three main perspectives in the data are considered: ontology, taxonomy, and IP (Intellectual Property). These provide a multi-perspective on the data involved in the lifecycle and serve as a basis for the proposed framework.

- Data ontology refers to the purpose of the data and its use; instead of presenting data as a raw material, it classifies the data by its purpose and use cases. In the case of maritime data uses, the most advanced ontology is presented in the SFI Coding and Classification System for Ship Information codification system, developed in the 1970s, *Manchinu and McConnell (1977).*

- Data taxonomy refers to the data standards and structure. Several commonly used file formats are available for data exchange in the ship design and construction industry, including IGES, STEP, IFC, JT, and OCX, among others. A separate group of data standards addresses data specifically in operations to enable IoT connections. The overall data structure highly depends on proprietary CAD vendor standards and often requires a costly integration interface to unlock the digital thread value with data transfers. A growing body of research in this area is usually disconnected from the shipyard's and ship design reality due to IP and commercial know-how restrictions.
- The IP refers to the ownership of the data; every data set or database has its owner, and often this ownership is not transferred to the next stage but remains with the creator, while only agreed-upon outcomes of the data are transferred to the subsequent phases. For example, if the basic design is performed in CAD software and in a 3D environment, it is often the case that the required outcome is based on 2D drawings, a specification, or an eBOM list. This creates dead-end routes in the digital thread, providing an area for improvement. The IP dimension also creates requirements for cybersecurity to protect the IP and prevent intentional or unintentional data manipulation.

3.1. Data in ship design, shipbuilding, and shipping

Each phase of the lifecycle produces specific data and refines data that flows into subsequent stages. Table I presents the most general categorization of phases, data generated, characteristics, outputs, and purpose.

Table I: Table captions the data ontology along typical shipbuilding phases in the lifecycle

Phase	Data Generated	Characteristics	Outputs	Purpose
Concept and Outline Specification	Principal dimensions, key performance requirements (speed, endurance, cargo capacity), fuel type, and environmental compliance strategy	High-level, scenario-driven, supporting conceptual decision-making	Outline Specification, preliminary GA plan, weight estimate, rough cost, and OPEX estimation	Provides the basis for preliminary quotations and initial owner–yard agreement
Basic Design	Hull form, compartment arrangements, major equipment specifications, propulsion and stability calculations, classification, and regulatory compliance results	Structured for class approval and regulatory submissions	Full Specification (Contract Specification), general arrangement plan, basic design drawings, and compliance documentation	Full Specification serves as the contractual baseline for shipbuilding
Detailed and Production Design	Structural parts, piping and cable routing, block division, detailed drawings	Massive expansion of data requiring integrated CAD/CAE/PLM systems	3D product models, working drawings, NC data, material lists, procurement specifications	Provides production-ready instructions consistent with the Full Specification
Procurement	Procurement specifications, RFQs, vendor proposals, purchase contracts, vendor drawings, type approval certificates,	Bridges design intent with supply chain capability, involving both technical and commercial data.	Approved vendor documents, purchase orders, vendor data packages	Ensures that equipment and systems meet design requirements and are delivered in

	test reports, and delivery schedules			time for construction
Construction Planning	Work breakdown structure (WBS), block division and assembly sequence, production schedules, integration of vendor delivery schedules with yard capacity	Synchronizes design and procurement outputs with shipyard resources	Construction schedule, work packages, production simulation results, resource plans	Optimize build efficiency, minimize bottlenecks, and ensure timely completion
Construction	Fabrication and installation records, inspection checklists, deviation reports, and updated as-built drawings.	Execution of production based on the construction plan; generation of as-built records	Construction records, quality assurance documentation, and updated as-built data.	Build the vessel as planned, ensuring both quality and adherence to schedule.
Commissioning and Trials	Functional and performance test results, system integration records, sea trial measurements	Validates compliance with the Full Specification and contractual guarantees	Trial reports, commissioning documentation, and verified as-built data.	Verify vessel performance against specification before delivery.
Operation and Maintenance (including Dry-Docking)	Operational logs, sensor and monitoring data, failure records, maintenance histories, dry-docking data	A combination of continuous operational data and periodic dry-dock inspection/repair records	Operational reports, predictive maintenance datasets, dry-docking reports, updated maintenance schedules	Operate safely and efficiently, maintain asset value, and feed back performance data
Decommissioning and Recycling	Material composition, residual life assessment, dismantling, and recycling plans	Safe dismantling, environmental compliance, and material recovery	Recycling documentation, environmental compliance certificates	Retire the vessel responsibly and sustainably

4. Digital Models, Twins, and Threads: types and purposes

Classification of data along the lifecycle can be based on various criteria, including multiple dimensions, types, or uses. A growing body of research on these topics presents a variety of perspectives and approaches. One of the emerging trends is the vertical–horizontal model for digital twin, developed by *Xiao et al (2022).* The model proposes combining real-time data with historical experience to visualize the evolution of digital twin models throughout their lifecycle. It accounts for the evolution of data throughout its lifecycle. It highlights the gaps between lifecycle stages as points where discontinuity in the digital thread hinders the benefits of digitalization. The following sections address the differences in these classifications and establish the vertical dimensions for the proposed framework.

4.1. 1D-2D-3D-4D and beyond data (N-D)

One-dimensional (1D) data in ship design primarily encompasses parametric specifications and linear measurements that form the foundational constraints for the development of vessels. For example, changing the dimensions of a ship's hull should automatically adjust related components such as

bulkheads and decks, ensuring that all parts fit together correctly.

These 1D parameters include overall length, beam, draft, and displacement ratios, which determine the basic vessel characteristics, as well as parameters for propulsion, energy efficiency, and other relevant factors. Two-dimensional (2D) data dominate traditional naval architecture through plan views, cross-sections, and technical drawings that represent complex three-dimensional forms on flat surfaces. Modern CAD systems aim to reduce the reliance on 2D drawings for manufacturing specifications, regulatory approval documentation, and construction blueprints by adopting 3D-based approaches. Still, often 2D formats remain essential for communication between design teams, classification societies, and shipyard workers who interpret these plans during fabrication processes. Additionally, 2D data is simple to generate and serves extremely well for sketching ideas; it becomes most useful when it is automatically converted or connected to the 3D data. 3D modeling has revolutionized ship design by enabling the complete virtual representation of a vessel before physical construction begins. The latest, most advanced approach incorporates time as the fourth dimension. In shipbuilding, 4D modelling integrates construction scheduling and project timeline management with spatial design data, allowing shipyards to visualize assembly sequences, identify potential conflicts, and optimize resource allocation throughout the build process. This temporal dimension enables predictive analysis of construction phases, material delivery scheduling, and workforce deployment, significantly reducing build times and costs. Advanced shipyards now utilize 4D digital twins that continuously update during construction, providing real-time monitoring of progress against planned schedules while enabling proactive problem-solving and quality control measures that ensure vessels meet both delivery deadlines and performance requirements.

Table II: An example of the classification of different data created and used along the lifecycle

Data dimension	**Examples of types of data**
1D	Specifications and classification rules Equipment lists Contracts and timelines
2D	General arrangement drawings Hull lines plan P&ID and system diagrams Deck and tank plans
3D	Hull shape forms Compartments arrangement 3D model of the complete design, including all disciplines
4D	Sequencing of assemblies and block construction Blocks erection planning Maintenance scheduling tied to the 3D models Ship lifecycle simulations: performance, voyage optimization

Novel research suggests adding a 5D dimension to accommodate costs along the 4D timeline, a 6D dimension to demonstrate sustainability and environmental performance, and a 7D dimension to incorporate the lifecycle operation and maintenance. Such classification can be considered derived data from the 3D data and represents a variety of digital twins for the selected purposes. Future trends indicate the emergence of the N-D: Digital Twin, which will consolidate real-time optimization of performance, routes, energy consumption, and logistics. Discussions also emerge about enabling AI-driven digital twins, where AI can connect multiple dimensions of data and generate predictive outcomes. This can also indicate the possibility of a universal digital twin, where data for each ship is consistently used throughout the lifecycle and for specific purposes at various stages.

4.2. Digital models and Digital Twins

The growing popularity of digital twin use and its applications is directly linked with digital transformation in the maritime industry. However, the terminology is often used for hype trends,

which don’t accurately correlate with the use of data. Digital twin implies the existence of both a digital model and a physical artifact, as well as an automatic connection between them. Based on the previously defined timeline, the digital twin begins to emerge near the end of the construction phase and will be most relevant during the operation phase. Both digital models and digital twins are often created for multiple connected purposes, such as simulating a physical property, for example, water resistance in CFD analysis, or energy efficiency for power generation/consumption. The universal multi-layered digital models often include a 3D model of a complete vessel and related 1D and/or 2D data at their core. In contrast, universal digital twins remain an idealistic target that the industry desires but has not yet made available, and which will require significant development in the standardization of data ontology and taxonomy, *Mauro and Kana (2023)*, *Fonseca and Gaspar (2021)*.

Connecting this concept with the previously discussed data dimensions, we can make a connection as follows. Digital models reside in the lifecycle stages before the end of the construction stage. At the same time, Digital Twins come into existence after the end of the construction stage, when the physical ship begins to emerge. Digital models primarily operate at 1D, 2D, and 3D levels, while Digital Twins take on levels four and beyond, and depending on the purpose, fall into one of the categories – 4D, 5D, 6D, or 7D. This representation aligns well with the high-dimensional space of LLMs (Large Language Models), and one can expect the emergence of a universal Digital Twin—an AI-managed Digital Twin with multidimensional data.

4.3. Digital threads and digital asset management

The disconnected use of data presents a significant challenge for the industry. The number of different 3D models used for various purposes in large projects can reach up to a hundred, and the narrow specialization of uses prevents the coherent use of data. This leads to extensive manual transfers and recreation of the data, which in turn causes errors and misalignments. PLM claims to tackle this problem, but remains on a level of promising a single source of truth and a single 3D model to include all possible use cases. Emerging AI capabilities can unlock significant gains in this area, considering the possibility of training a dedicated SLM (Small Language Model) for the industry's specific purposes and the potential use of AI to orchestrate the data and APIs of authoring applications.

Digital asset management has the potential to address data usage coherence throughout its lifecycle. Several critical implications must be considered to achieve the desired results of the approach. First of all, this is not exclusively about IT or software solutions. A combination of data, digital twin technologies, and cooperation between engineering and operations is required, *Hideyuki, (2020)*. For example, ship owners may lack the necessary skills to update 3D models for CFD analysis. Therefore, to utilize operational-phase data in ship design as an operational profile, cooperation is needed from design companies and/or the design departments of shipyards that possess the required expertise. This makes a transition more complex than just a data transfer, but also opens possibilities for design companies to provide services. Another concern is the data formats, as such data is typically stored in specialized applications, such as CAE/CAD/PLM or similar. The lifecycle of these applications generally spans 3 to 5 years, as technology necessitates updates to hardware and/or software. This significantly shorter time period compared to the ship lifecycle; therefore, it is often recommended to resort to simple and established data formats, which in turn hinders the possibilities of state-of-the-art data taxonomies to address the complex nature of shipbuilding data, such as topology and interconnections.

5. Framework for data lifecycle and data maturity

The proposed framework, Fig.4, aims to provide a visual representation of the data throughout the ship lifecycle. It combines the discussion from the first part about the timeline of the ship lifecycle and the main points of the debate regarding the data, its dimensions, ownership, and maturity.
A horizontal timeline presents lifecycle stages grouped by the main stages: ship design, shipbuilding,

and shipping. The stages are general or indicative, highlighting the main concepts, and can be adjusted as needed. Three different cases are presented on the vertical scale: data dimensions, data maturity, and ownership:

- The first case highlights the level of maturity of data. In the initial stages of the lifecycle, the data is uncertain and undergoes iterations within the design spiral. However, as it progresses to the stages of basic design, it gains maturity, a process that continues until the end of the lifecycle. An example of such data is hull form and main dimensions, which are defined and verified through calculations at the early stages of design and remain static later on. The same happens with the 3D data; once the model is finalized, it remains helpful in the later stages of the timeline. After the commissioning, the data used for construction is no longer needed to the same extent, while the maturity of operational data increases.
- The second case presents the differences of the data dimension created, from 1D/2D information in the initial stages – specification, GA, cost estimations, etc, to the 3D models in detailed design stages, complemented with planning data and construction scheduling (4D). At the operational stage, there may be potentially multiple mode dimensions, such as 5D-7D, using this terminology.
- The third case outlines the ownership of the data, which is shown as four boxed areas that correspond to naval architecture (or functional design), ship design, shipbuilding, and operations.

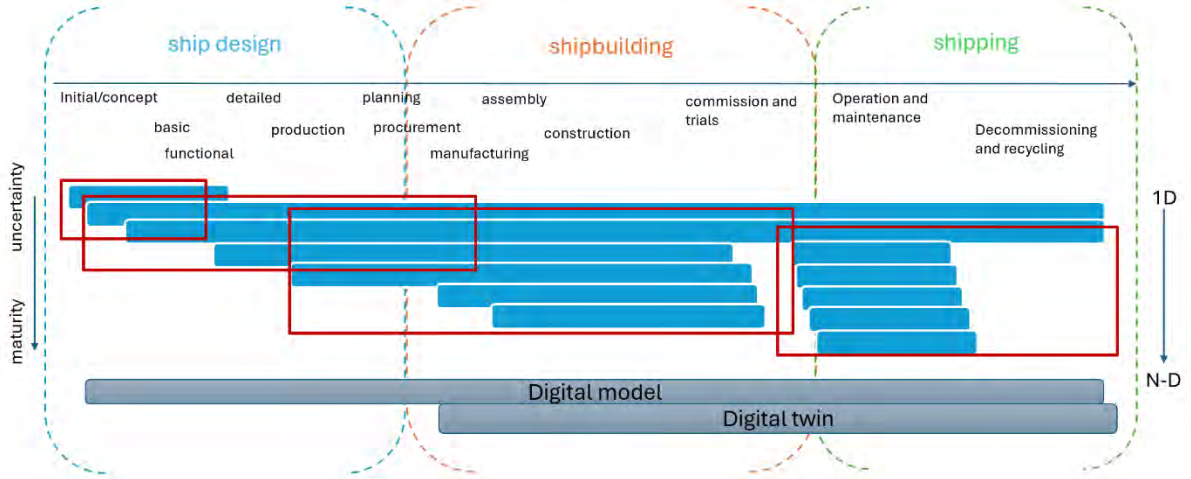


Fig.4: Visual framework for ship lifecycle and data maturity, dimensions, and ownership. The horizontal axis represents a timeline of ship lifecycles, while the vertical axes indicate data maturity as it increases at each stage and data dimensions. Red boxed areas indicate typical ownership of data for naval architecture, ship design, shipyards, and shipowners.

This framework visualizes the complexity behind data, illustrating that the common assumption that the amount of data increases over time in the project and gradually builds up is misleading. The following conclusions can be made based on the presented framework:

- The main transition points are between the main stages, and these can be observed in data dimensions, level of maturity, and ownership
  - These transitions are related to the historical split of labour, such as between engineering and construction, but present a significant difficulty in establishing a continuity of data thread and enabling the transition between naval architecture, design, building, and operations. Focusing on these transition points can leverage the use of data and improve the outcomes of digitalization efforts.
  - Lack of the concept of digital asset management deprives the industry from benefiting from the digital transformation. If outputs are valued as contractual issues, the 3D data could be included as a digital asset and managed in further lifecycle stages.

- Digital asset management can elevate the data usage and facilitate the digital thread for the entire life cycle; however, it requires a mind shift for the main stakeholders, standardization of data ontology and taxonomy, ownership and IP, and support for long lifecycle from the software solutions.
- There is a visible data backbone that extends from the end of the basic design stage to the end of the lifecycle. This backbone contains mostly 1D/2D/3D data, which defines` the main parameters of the ship, such as hull form and compartment arrangements (3D), GA, P&IDs, and other agreement drawings(2D), and a substantial amount of 1D data, such as reference dimensions, stability, and loading conditions calculations, etc
  - Growing interest in this data to be transferred to the 3D formats is reflected in the 3DMBA process
  - It can be argued further that, in some cases, this data is efficiently transferred and used by ship owners and ship operators; however, such data transfers almost always focus on the handover of documentation instead of a complete digital asset
- Emergent new technology, such as AI, can change the industry drastically, for example, from the perspective of N-D or AI-managed Digital Twin. For the foundation of such data models, the data ontology should be methodologically solid, and the data taxonomy should support the newest technology, such as MCP (Model Context Protocol), to connect different proprietary applications with an AI-managed data governance.

5.1. Example of data flows through the lifecycle

To provide context for the proposed framework, we can use two examples: the basic design of the vessel and its electrical and outfitting systems, which involve a power generator, as well as how related data evolves during the vessel's lifecycle in these cases.

For basic design, the principal dimension is typically 1D data, which originates at the earliest stage of the lifecycle with high uncertainty values. Through subsequent design spirals, initial 3D data is generated by progressing through steps such as cargo capacity, compartmentation & tank arrangement, displacement/weight, stability, propulsive performance, general arrangement, machinery arrangement, hull form/structure, outfitting & systems, regulations, manufacturability, and cost estimation. The 3D data for the vessel as a platform becomes fixed and static from the stage of basic design forward. These are highly unlikely to be changed; at most, these might be slightly updated or altered in the retrofit projects later in the operational stage. This type of data is created and verified by naval architects and will form part of the backbone that spans the entire lifecycle. It would be beneficial to integrate this data as an integral part of the hull form and various specialized digital models and twins, ensuring that it is automatically “inherited” from previous stages or is part of the digital asset.

Another example is electrical and outfitting systems involving power generators, which are typically defined as one of the main pieces of equipment on board in the ship's specification. There, power demand according to the main engine output and the operating phase is summarized in an electrical load balance table. Additionally, generator specifications, including dimensions, weight, fuel consumption, manufacturer, and model, are defined. All of this information is 1D data. At the basic design stage, electrical systems are represented using single-line diagrams, and outfitting systems are depicted using P&I diagrams with 2D symbol representations for this equipment. GA will provide an estimated location for it on the ship. On the detailed design stage, it will become a 3D component with a volume and connections for piping, electricity, and other systems. For the assembly stage, it will receive a 4D layer of data specifying the time of delivery and installation on board, as well as 5D, 6D, and 7D data, including costs and other relevant layers of information. Once in operation, additional data will begin to appear, collecting IoT and sensor data to inform performance and maintenance predictions. For the decommissioning stage, it would be useful to have the original data from manufacturing and maintenance for defining whether any parts can be recycled. In this example, the data gradually builds up on the dimensions layers. Still, there is a distinctive split between building and operation stages, while the primary base data is used on all stages.

6. Conclusions

The shipbuilding and shipping industries are at a pivotal moment in their digital transformation journey. Despite the growing availability of advanced tools and methodologies, the fragmented nature of data across lifecycle stages continues to hinder the full realization of digital asset management and integrated decision-making. This paper has examined the lifecycle of ships through both horizontal and vertical lenses—highlighting the timeline of design, construction, and operation, as well as the evolving dimensions, maturity, and ownership of data.

By comparing maritime practices with those in more digitally mature industries, and by analyzing the roles of digital models and digital twins, we have identified critical gaps and opportunities for improvement. The proposed framework offers a structured approach to understanding and managing data transitions, enabling stakeholders to better align engineering, operational, and business objectives. To unlock the full potential of digital transformation, the industry must adopt standardized data ontologies and taxonomies, redefine data ownership and model ownership, and ensure long-term data continuity. Emerging technologies, such as the use of LLM/SLM and AI-managed digital twins, present promising avenues for overcoming current limitations. However, their success depends on a solid foundation of interoperable data taxonomy and collaborative data ontology practices across the value chain. The desired shift toward coherent digital asset management is not just a technical challenge—it is a strategic imperative for improving lifecycle efficiency, sustainability, and competitiveness in the maritime domain.

References

BRETT, P.O.; GARCIA AGIS, J.J.; LAGEMANN, B. (2024). *Early Marine Systems' Design – Cracking the wicked problem - The case of a novel biomass harvesting vessel,* Int. Marine Design Conf.

BRUCE, G. (2021) *Shipbuilding Management*, Springer Singapore

EIGNER, M. (2021), *System Lifecycle Management, Engineering Digitalization,* Springer

EVANS, H.J. (1959), *Basic Design Concepts,* ASNE https://doi.org/10.1111/j.1559-3584.1959.tb01836.x

FONSECA, I.A.; GASPAR, H.M. (2021), *Challenges when creating a cohesive digital twin ship: A data modelling perspective*, Ship Technology Research 68/2, pp.70-83

HIDEYUKI, A. (2020) *Digitalization in the Maritime Industry*, ClassNK Technical J. 1, pp.5-12, https://www.classnk.or.jp/hp/pdf/research/rd/giho01e_2019.pdf

HITCHCOCK, J. (2021), *Shipbuilding leadership: financial responsibility, operational models & team integration,* Independent publisher

MANCHINU, A.; McCONNELL, F. (1977), *The SFI Coding and Classification System for Ship Information*, REAPS Technical Symp., Louisiana

MAURO, F.; KANA, A. (2023), *Digital twin for ship life-cycle: A critical systematic review*, Ocean Eng. 269, https://doi.org/10.1016/j.oceaneng.2022.113479

PAPANIKOLAOU, A. (2014), *Ship Design, Methodologies of Preliminary Design,* Springer

XIAO, W.; HE, M.; WEI, Z.; WANG, N. (2022), *SWLC-DT: An Architecture for Ship Whole Life Cycle Digital Twin Based on Vertical–Horizontal Design*, Machines 2022, 10, 998, https://doi.org/10.3390/machines10110998

From MINLP to QUBO: A Quantum-Inspired Approach for Bulk Carrier Routing with Time Windows

Joshua Dibbern, Fraunhofer CML, Hamburg/Germany, joshua.dibbern@cml.fraunhofer.de
Oliver Szal, Fraunhofer CML, Hamburg/Germany, oliver.szal@cml.fraunhofer.de
Anisa Rizvanolli, Fraunhofer CML, Hamburg/Germany, anisa.rizvanolli@cml.fraunhofer.de

Abstract

Emerging optimization paradigms like Quantum Annealing (QA) offer potential for tackling large-scale combinatorial problems, yet applications in the shipping industry remain scarce. This paper addresses the cargo routing problem for bulk carriers, where fuel consumption depends nonlinearly on vessel speed and strict time windows must be respected. We formulate the problem as a Mixed-Integer Nonlinear Program (MINLP) and derive a tractable model through speed discretization, yielding a Travelling Salesman Problem with Time Windows (TSPTW). A Quadratic Unconstrained Binary Optimization (QUBO) model enables exploration with QA, and performance is evaluated using classical solvers and Simulated Annealing (SA).

1. Introduction

Maritime transportation plays a crucial role in the backbone of global trade, with bulk carriers accounting for a substantial fraction of transported goods such as coal, iron ore, and grain. Since 1970, international maritime trade with bulk carriers has risen from 448 million tons to 3,272 million tons in 2021, *Christiansen et al. (2025)*. Even though the margin for transportation is usually high, the operation of vessels is in general strongly cost-driven. Fuel consumption constitutes one of the dominant components of total voyage expenses and is tightly linked to both routing decisions and vessel speed. Furthermore, the ecological significance of reducing fuel consumption must also be mentioned. In 2019, bulk carriers caused 250 million tons of CO_2 emissions, with individual bulk carriers averaging 15,000 tons CO_2, *Sirimanne et al. (2020)*. Therefore, even marginal improvements in operational efficiency can yield significant financial and environmental benefits, considering the scale of worldwide bulk shipping. However, identifying the optimal combination of routes and speeds becomes intractable for classical methods as the number of possible travel routes grows exponentially with each additional port call. This motivates the exploration of novel computational paradigms such as quantum computing, which leverages quantum effects like superposition, entanglement and tunnelling to explore vast solution spaces in a probabilistic way and could enable real-time route optimization at global scale in the future. In this paper we show first attempts to bring together these two different worlds and take the first steps toward investigation of quantum computing potentials for the maritime domain by formulating the problem in a way that state-of-the-art quantum computers can handle.

In collaboration with the German shipping company Harren Group specialized in integrated project logistics and heavy-lift transport, we address a real-world operational challenge. Their bulk and breakbulk fleet ranges from Mini Bulkers to Post-Panamax vessels, with capacities between 36 000 and 92 000 dwt, serving diverse trade lanes. Unlike container shipping, bulk operations face notable demand volatility alongside a handful of recurring routes. Meeting this dynamic environment demands rapid optimization and continuous adaptation of voyage plans. Route planning must balance numerous variables - customer requirements, weather, port and bunker costs, etc. - which makes manual, experience-driven iteration both time-consuming and inflexible. A fully integrated, system-based optimization that considers all these factors often struggles to deliver targeted solutions, because the underlying optimization problems are complex and computationally intensive. Quantum computing has the potential to provide significant advantages for solving such large-scale combinatorial optimization problems in the future, but the currently limited capacity of available hardware limits the size of instances which can be solved. To enable a meaningful investigation of quantum computing potentials, we therefore concentrate on a reduced problem setting that highlights the most impactful aspects.

In this context, we formulate the cargo routing problem (CRP), which addresses the question of how a bulk carrier should visit a set of ports to minimize fuel consumption, as the main part of the operational costs, while respecting contractual obligations. The problem is characterized by a depot (home port), a set of ports that must be visited exactly once, and strict time windows specifying earliest and latest possible service times at each port. These constraints arise due to port availability, contractual delivery times, and tidal restrictions. In addition to routing decisions, the vessel operator must also determine the optimal cruising speed, since fuel consumption grows nonlinearly with speed, *Schneekluth and Bertram (1998)*. To keep the model tractable, we assume a constant cruising speed throughout the voyage. Thus, the overall objective in the cargo routing problem is to jointly optimize the sequence of visited ports and the speed at which the vessel travels between them in order to minimize total fuel cost. In the following sections it is shown that this leads to a Mixed Integer Nonlinear Problem (MINLP), which is computationally expensive to solve. However, by discretizing the speed, the problem can be reformulated as the well-known Travelling Salesman Problem with Time Windows (TSPTW), a combinatorial NP-hard optimization problem that generalizes the classical TSP.

Classical exact methods, such as Branch and Cut for Mixed-Integer Linear Programming (MILP) formulations, can efficiently solve some instances of the TSPTW but scale poorly when dealing with large real-world instances due to the exponential growth of the search space. Consequently, heuristic and metaheuristic approaches, including Simulated Annealing (SA), have been employed to obtain high-quality solutions within reasonable computation times, *Ohlmann and Thomas (2007)*. SA is inspired by the physical process of annealing in metallurgy and explores the solution space probabilistically by doing random variable flips to escape local optima, *Kirkpatrick et al. (1983).*

More recently, advances in quantum computing have given rise to Quantum Annealing (QA), a probabilistic optimization paradigm that encodes combinatorial problems into Quadratic Unconstrained Binary Optimization (QUBO) models and solves them using the adiabatic principle and quantum tunnelling effects, *McGeoch (2014), Lucas (2014)*. For that, a system of qubits is prepared in the ground state of a well-known Hamiltonian, e.g. in an equal superposition of all qubits. Then the system is adiabatically driven to the desired ground state (encoding the optimal solution of the QUBO) by slowly changing the energy to a Hamiltonian whose interaction-terms correspond to the QUBO-coefficients. Special purpose hardware, e.g. D-Wave quantum annealers, directly implement this paradigm in their hardware topology. QA provides a natural platform for many optimization problems like the TSPTW, which can be mapped into QUBO formulation, *Papalitsas et al. (2019), Salehi et al. (2022)*. The potential advantage lies in exploiting quantum effects to explore the vast solution space more effectively, possibly outperforming classical heuristics for certain structured instances.

The focus of this work is to formulate the cargo routing problem for a single bulk carrier as a TSPTW in MILP and QUBO form. This allows for a performance comparison of classical solvers and simulated annealing on small but representative problem instances. The overall motivation is twofold: First, to demonstrate how optimization models can directly translate into tangible fuel savings in maritime operations, and second, to pave the way for evaluating quantum optimization methods for potential large-scale applications in the future of shipping logistics.

2. Problem description

This section shows how the cargo routing problem, i.e. minimizing the fuel costs of a cargo ship by varying the route and speed, can be reduced to a TSP with time windows. This work focuses on the first investigations towards using quantum computing approaches for solving maritime transportation problems. Therefore, when it comes to fuel consumption models for cargo ships, we consider simplified models as the “cube law”, according to which a ship’s power requirement grows roughly with the cube of its speed. We are aware of the fact that this traditional resistance model is very simplified and works best for design speed. Fuel use is affected by far more nonlinear and stochastic factors like draft, trim, hull fouling, wave and wind conditions, salinity, depth, temperature, etc. Numerous fuel-consumption functions have been proposed with more accurate consideration of these effects, *Schneekluth and Bertram (1998), Newman (2018).* The various models also identify speed as the main driver - through

required horsepower and specific fuel consumption - even though its relative importance shifts with weather-induced resistance. Despite these developments, for this work we will consider the simplified model as a surrogate representation of the nonlinearities when considering fuel consumption. Furthermore, we make the simplifying assumption that the speed is constant through the whole tour.

2.1. Fuel model

From the standard hydrodynamic-based assumption (known as "propeller law") it can be deduced that the required propulsion power P of a ship can be approximated by the formula

$$P \approx k_1 V^3,$$

where V is the speed of the ship and $k_1 > 0$ is a ship-dependent coefficient that depends, among other things, on the diameter of the propeller, *Newman (2018).* Hence the fuel F consumed during a voyage of duration t, length s and (constant) speed V can be approximated by

$$F \approx k_2\, t\, P = k_2\, \frac{s}{V}\, k_1 V^3 \;=\; k_1 k_2\, s\, V^2$$

for an engine-dependent coefficient $k_2 > 0$. Defining $C \coloneqq k_1 k_2 > 0$, we obtain

$$F(s, V) \approx C\, s\, V^2,$$

which means that the fuel consumption depends linearly on the length of the tour s and quadratic on the ship speed V.

2.2. Derived problem setting

To minimize the fuel consumption F, we consider now the following problem setting which results in a nonlinear model and is inspired by the TSPTW formulation of *Hungerländer and Truden (2018)* and *Kara and Derya (2015).* The problem concerns a single bulk carrier that is instructed to visit a finite set of ports (or cities, customers) $\mathcal{C}$, starting and ending at a home port (depot). Each port $u \in \mathcal{C}$ needs to be serviced within a prescribed time window $[e_u, l_u] \subset \mathbb{R}_+$. Travel between ports is characterized by a symmetric distance $c_{uv} > 0$ for $u, v \in \mathcal{C}$ with $u \neq v$, which can be interpreted as travel times or travel costs. The idea is to view the ports as vertices in a fully connected graph, where the symmetric weights of the edges correspond to the travel distances (or travel times) between the cities.

The considered optimization problem is to choose a Hamiltonian cycle (i.e. a tour which visits each port exactly once, starting and ending at the depot) and a constant speed V for the entire tour such that the total fuel consumption is minimized while all time windows are respected. To enable this, each edge is equipped with a binary decision variable to determine if a direct route between two cities is part of the tour or not. The speed is treated as a single, continuous decision variable which influences the travel times and has a strong effect on fuel consumption.

Sets and indices

- depot: 0 (start) and auxiliary sink node $n + 1$ (end), which represents the depot as well,
- ports/customers/cities: $\mathcal{C} = \{1, \dots, n\}$ for an $n \in \mathbb{N}_{\geq 2}$,
- abbreviate $\mathcal{C}_0 \coloneqq \mathcal{C} \cup \{0\}$ and $\mathcal{C}_{n+1} \coloneqq \mathcal{C} \cup \{n+1\}$,
- let $\mathcal{C}^2 \coloneqq \{(u, v) \text{ with } u, v \in \mathcal{C} \text{ and } u \neq v\}$ and analogously $\mathcal{C}^3$ (for tuples of 3 ports which are pairwise different).

Parameters

$c_{uv} \in \mathbb{R}_{>0}$	travel cost $u \in \mathcal{C}_0$ to $v \in \mathcal{C}_0 \setminus \{u\}$,
$e_u \in \mathbb{R}_{\geq 0}$, $l_u \in \mathbb{R}_{\geq 0}$	earliest and latest allowed service times at port $u \in \mathcal{C}$,

$V_{\min}, V_{\max} \in \mathbb{R}_{>0}$, with $V_{\min} < V_{max}$ lower/upper bounds on the constant route speed,
$C \in \mathbb{R}_{>0}$ constant for ship and engine parameters (fuel model).

Decision Variables

$x_{u,v} \in \{0,1\}$ $\forall (u,v) \in \mathcal{C}^2$, 1 iff the tour visits v immediately after u,
$x_{0,v} \in \{0,1\}$ $\forall v \in \mathcal{C}$, 1 iff the tour leaves depot 0 to v,
$x_{u,n+1} \in \{0,1\}$ $\forall u \in \mathcal{C}$, 1 iff the tour leaves u to return to depot $(n+1)$,
$A_u \in \mathbb{R}_{>0}$ $\forall u \in \mathcal{C}_{n+1}$, arrival time at port u,
$V \in [V_{\min}, V_{\max}]$, constant sailing speed used on the entire tour.

Thus, the number of continuous decision variables is $n+2$ and the number of binary variables is

$$|\mathcal{C}^2| + 2|\mathcal{C}| = n^2 + n.$$

Objective (nonlinear)

$$\min_{x_{u,v}, V} C\left(\sum_{(u,v)\in\mathcal{C}^2} c_{uv}\, x_{u,v} + \sum_{v\in\mathcal{C}} c_{0v}\, x_{0,v} + \sum_{u\in\mathcal{C}} c_{u0}\, x_{u,n+1}\right) V^2 \quad .$$

Constraints

(1) each port entered exactly once:

$$\sum_{u\in\mathcal{C}_0,\, u\neq v} x_{u,v} = 1 \quad \forall v \in \mathcal{C},$$

(2) depot $(n+1)$ entered exactly once:

$$\sum_{u\in\mathcal{C}} x_{u,n+1} = 1,$$

(3) each port left exactly once:

$$\sum_{v\in\mathcal{C}_{n+1}, v\neq u} x_{u,v} = 1 \quad \forall u \in \mathcal{C},$$

(4) depot (0) left exactly once:

$$\sum_{v\in\mathcal{C}} x_{0,v} = 1,$$

(5) time-window compliance:

$$e_u \le A_u \le l_u \quad \forall u \in \mathcal{C},$$

(6) arrival-travel relations:

$$A_u - A_v + \left(M_{uv} + \frac{c_{uv}}{V}\right) x_{u,v} \le M_{uv} \quad \forall (u,v) \in \mathcal{C}^2,$$

(7) arrival after leaving depot:

$$\frac{c_{0v}}{V} x_{0,v} \le \mathrm{A_v} \quad \forall v \in \mathcal{C},$$

(8) arrival when return to depot:

$$A_u + \frac{c_{u0}}{V} x_{u,n+1} \le A_{n+1} \quad \forall u \in \mathcal{C}.$$

A convenient tailored choice is $M_{uv} := l_u - e_v$ for $(u,v) \in \mathcal{C}^2$, which makes constraint (6) active when $x_{uv} = 1$ and relaxed otherwise. In practice one must ensure $M_{uv} \ge 0$ for all pairs which can be active; pairs (u,v) with $l_u < e_v$ are infeasible a priori and can be removed from consideration.

Remarks on the model

- This model is a Mixed Integer Nonlinear Program (MINLP) with nonlinearities at two places: The objective is nonlinear (cubic) and the constraints (6)-(8) are inversely proportional in V.
- Additional explicit subtour elimination constraints are not required. The arrival time variables A_u, together with the time-window restrictions, implicitly enforce an ordering of the visited cities and

thereby rule out subtours that do not include the depot. This mechanism is conceptually similar to the well-known Miller–Tucker–Zemlin (MTZ) constraints, which are often used to formulate TSP.

- W.l.o.g., service times are omitted here to simplify the notation and avoid introducing additional variables. They can be integrated into the arc travel times c_{uv}.
- To make this model more general one could also relax the assumption of the constant speed V for the whole tour and require a constant speed $V_{u,v}$ for each travelled arc $(u, v) \in \mathcal{C}^2$. The overall structure of the given model would remain unchanged, only the number of the continuous decision variables, which affect the model in a nonlinear way, would increase by $n(n-1)+n+n-1$ variables, which makes the problem even harder.

2.3. Discretizing the speed

The above MINLP is nonconvex, since the objective involves the quadratic term V^2 and the time-propagation constraints involve terms of the form $\frac{1}{V}$. Solving such models exactly is computationally expensive and scales poorly with the number of ports. A common approach to address this challenge is to discretize the continuous speed interval (for $m \in \mathbb{N}$)

$$[V_{\min}, V_{\max}] \rightarrow V_m = \{V^{(1)}, \dots, V^{(m)}\}.$$

Then, for each fixed speed $V^{(k)}$ one obtains a (separate) Mixed Integer Linear Program (MILP) since the travel times $\frac{c_{uv}}{V^{(k)}}$ become constants and the objective becomes linear. This yields a tractable model that can be solved with modern branch-and-cut solvers such as CPLEX.

Because there could be a MILP with no existing feasible routing solution for a given $V^{(k)}$, one should start with the largest $V^{(k)}$. If one of the MILPs has no feasible solution, smaller speeds can be ignored and while only greater speeds (up to $V_{\max}$) are considered.

Denote by $x^{(k)}$ the optimal solution of the MILP which is associated to $V^{(k)}$, and by $L\left(x^{(k)}\right)$ its total distance. After computing all solutions $x^{(k)}$ the best pair (route, speed) is chosen by comparing the fuel objective values

$$F^{(k)} = C \cdot L\left(x^{(k)}\right) \cdot \left(V^{(k)}\right)^2$$

over $k = 1, \dots, m$. A natural discrete approximation of the original MINLP solution is the pair $\left(x^{(k^*)}, V^{(k^*)}\right)$, where

$$k^* \in \arg\min_{k=1,\dots,m} F^{(k)}.$$

This discretization yields a tractable pipeline that is straightforward to implement, parallelizes trivially across k, and produces a controllable approximation of the original MINLP.

Alternative options to tackle the MINLP include nonlinear global optimization solvers (e.g. Couenne, *Belotti et al. (2009),* or BARON, *Zhang and Sahinidis (2024)*), which can provide exact solutions but typically do not scale well, and heuristic or metaheuristic approaches (e.g., local search, evolutionary methods, *Young et al. (2007)*), which can scale better but do not provide guarantees of optimality. Because the speed discretization strikes a practical balance between accuracy and solvability for realistic cargo routing problems in maritime operations, it is chosen here as an ansatz to solve the problem.

3. Methods

3.1. QUBO

A Quadratic Unconstrained Binary Optimization Problem (QUBO) is a combinatorial optimization problem of the form

$$\min_{x\in\{0,1\}^n} x^\top Q x \;=\; \min_{x_i\in\{0,1\},\, i=1,\dots,n} \sum_{i,j=1}^{n} Q_{ij}\, x_i x_j\,,$$

$Q = \left(Q_{ij}\right) \in R^{n\times n}$ is a symmetric real-valued matrix of coefficients and the vector $x = (x_i) \in \{0,1\}^n$ encodes the binary decision variables. The QUBO formulation is *unconstrained* in the sense that it does not explicitly contain constraints. Instead, feasibility requirements of the original problem are typically encoded into the quadratic objective using *penalty terms*. For example, constraints of the form $\sum_{i=1}^{n} x_i = 1$, with binary variables x_i, can be incorporated into the objective function by adding a penalty term

$$p\left(\sum_{i=1}^{n} x_i - 1\right)^2 = p\left(\sum_{i,j=1}^{n} x_i\, x_j - 2\sum_{i}^{n} x_i + 1\right) = p\left(\sum_{i,j=1}^{n} \widetilde{Q_{ij}}\; x_i x_j\right) + p,$$

where $p > 0$ is a sufficiently large penalty weight and the $\widetilde{Q_{ij}}$ are chosen as -1 for $i = j$ and as 1 else. In this way, constrained optimization problems can be transformed into equivalent unconstrained quadratic formulations. In case of inequality constraints, e.g. $\sum_{i=1}^{n} a_i y_i \le b$ with integer variables y_i, binary slack variables S_k must be added to the left side to turn it into an equality constraint

$$\sum_{i=1}^{n} a_i y_i + \sum_{k=0}^{K} 2^k S_k = b.$$

The amount of required slack variables depends on the specific constraint. K should be chosen large enough, such that the sum $\sum_{k=0}^{K} 2^k S_k$ is greater equal than $b - \min_{y} \sum_{i=1}^{n} a_i y_i$.

A critical aspect of any QUBO formulation is the choice of penalty parameters p. If the penalties are too small, the solver may return infeasible solutions that violate the original constraints. If the penalties are excessively large, the feasible region is enforced but the cost terms Q_C are dwarfed, leading to poor energy scaling, numerical instability, and reduced solution quality on both classical and quantum hardware. In practice, penalty parameters must be chosen carefully to balance constraint enforcement and cost optimization. This often requires empirical tuning or problem-specific heuristics, *Lucas (2014)*. In our experiments, we explicitly varied the penalty weights and analyzed their impact on the quality of feasible solutions, as will be shown in the results.

QUBO models are of central importance because they constitute the standard input format for many modern heuristic and quantum optimization algorithms, such as simulated annealing, quantum approximate optimization algorithm (QAOA), and in particular quantum annealing as implemented on hardware platforms like the D-Wave quantum computer. Furthermore, there exists a one-to-one correspondence between QUBO problems and Ising models from statistical physics. In summary, the QUBO formulation provides a generic and flexible mathematical framework for encoding a wide variety of combinatorial optimization problems into a common structure. Its unconstrained quadratic form enables direct applicability of both classical metaheuristics and emerging quantum computing methods.

3.2. Simulated Annealing

Simulated annealing is a classical sample-based optimization heuristic that fits into the class of Monte Carlo methods, *Kirkpatrick et al. (1983)*. Throughout the runtime of a simulated annealing run an inverse temperature parameter $\beta = 1/(k_B T)$ is set and gradually decreased. T and k_B correspond to the temperature and Boltzmann constant of a Boltzmann distribution. Samples of variable values are generated according to some random distribution. Then for each value of β and for every variable a variable flip is proposed by comparing the energies (objective functions) in the initial and flipped case. Whether a variable flip is accepted depends both on the energy difference and the current inverse temperature β. In particular, higher temperatures (smaller values of β) increase the likelihood of a variable flip even in cases where the flip would increase the overall energy. This makes it possible to escape local minima. With advanced runtime, as the system "cools down", the variable flips towards higher energies become less likely, such that ideally the system settles close to the global minimum. This process is repeated

for a variable number of annealing runs to obtain a batch of random solution samples. For our benchmark we used the D-Wave simulated annealing implementation given by the dwaveneal Python library, https:/github.com/dwavesystems/dwave-neal.

4. QUBO formulation of TSPTW

The Travelling Salesman Problem with Time Windows (TSPTW) has been extensively studied in the operations research literature. Classical formulations typically extend the Miller-Tucker-Zemlin (MTZ) model or employ flow-based formulations with additional time window constraints, *Desrosiers et al. (1995), Dumas et al. (1995).* Due to its NP-hardness, exact branch-and-bound and branch-and-cut methods scale poorly with the number of customers, so that for a long time the practical solvability was limited to instances with up to a few dozen nodes. Consequently, heuristic and metaheuristic algorithms such as tabu search, *Ho and Haugland (2004),* genetic algorithms, *Ohlmann and Thomas (2007),* ant colony optimization, *Gambardella et al. (1999),* and simulated annealing, *Osman (1993),* have been widely applied to obtain near-optimal solutions for larger instances.

With the emergence of quantum computing, the TSP and its variants have increasingly been investigated in the context of QUBO formulations to establish quantum algorithms like quantum annealing as solution methods. The edge-based QUBO formulations of TSP and TSPTW, as discussed in *Papalitsas et al. (2019)* or *Salehi et al. (2022)*, provide a systematic framework to translate combinatorial routing problems into a binary quadratic form. Early computational experiments on D-Wave quantum annealers, *Venturelli et al. (2015), Neukart et al. (2017),* demonstrated the feasibility of solving small TSP instances, although scaling and embedding remain major challenges. More recent works focus on hybrid approaches, combining classical preprocessing with quantum annealing to handle time windows and resource constraints, *Lucas (2014), Feld et al. (2019).*

In the maritime context, routing and scheduling problems under time constraints have been studied primarily with classical optimization techniques, with only very limited attention to quantum formulations so far*, Masuda et al. (2023)*, *Szal et al. (2025).* This gap motivates the present work: to systematically compare a classical MILP formulation of the TSPTW for bulk carrier routing with its QUBO counterpart, thereby providing one of the first case studies where a maritime routing problem is explicitly reformulated in a way that makes it compatible with quantum annealing hardware.

We begin by stating that the MINLP model from section 2 can be converted into a MILP by two significant changes, thereby eliminating the dependence on speed V: First, the V^2 is omitted from the objective and second, in the arrival-travel relations (constraint (6)), the term $\frac{c_{u,v}}{V}$ is replaced by $c_{u,v}$. The resulting MILP represents the classical side of our computational studies.

Now a QUBO formulation for the symmetric TSPTW is given, which follows the work of *Salehi et al. (2022)* and *Papalitsas et al. (2019)*. Like the MINLP/MILP models given earlier (in section 2) the ansatz is edge-based, meaning that the decision variables encode a travelled arc. But unlike the MINLP/MILP models, the decision variables are also equipped with another index, which represents discrete time steps.

In *Salehi et al. (2022)* two alternative QUBO formulations of the TSPTW are mentioned: a node-based one that describes whether a city occupies a certain position in the tour, and a special edge-based, derived from an ILP approach. While the node-based formulation originally requires the fewest variables, it is a HUBO that contains higher order terms, which need to be quadratized first before being able to solve by annealers. This leads to an overhead, possibly eradicating the advantage of the native fewer additional variables and constraints, *Salehi et al. (2022)*. The ILP approach starts with few variables but requires many constraints such that after conversion to QUBO we end up with more variables than in the edge-based case. Therefore, and due to its conceptual closeness to the classical MILP model we chose the edge-based ansatz.

Sets and indices

Complementary to the previously given sets and indices in section 2, time steps $i = 1, \dots, n+1$ are needed. Since the time steps 1 and $n+1$ for the edge (travelled arc) starting rsp. ending at the depot are treated differently in the constraints, we define $I \coloneqq \{2, \dots, n\}$.

Parameters

For simplicity, we assume in the following that the parameters c_{uv}, e_u, l_u from earlier are integers.

Decision variables

$x^{\mathrm{i}}_{u,v} \in \{0,1\}$ $\quad \forall (u,v) \in \mathcal{C}^2,\ i \in I,$ $\quad$ 1 iff u and v are at positions $i-1$ and i of the tour,

$x^{\mathrm{i}}_{0,v} \in \{0,1\}$ $\quad \forall v \in \mathcal{C},$ $\quad$ 1 iff customer v is the first customer of the tour,

$x^{\mathrm{i}}_{u,0} \in \{0,1\}$ $\quad \forall u \in \mathcal{C},$ $\quad$ 1 iff customer u is the last customer of the tour.

Thus, the amount of binary variables is

$$s \coloneqq |I||\mathcal{C}^2| + 2|\mathcal{C}| = n(n-1)^2 + 2n.$$

Objective (preliminary)

First, we state the base objective containing only the travel costs, which will be enriched by the penalty terms from the constraints in the next steps

$$\min_{x \in \{0,1\}^s} \sum_{(u,v) \in \mathcal{C}^2} c_{uv}\ x^i_{u,v} + \sum_{v \in \mathcal{C}} c_{0v}\ x^1_{0,v} + \sum_{u \in \mathcal{C}} c_{u0}\ x^{n+1}_{u,0}.$$

This linear objective can easily be transformed into a QUBO problem. Denote the corresponding quadratic coefficient matrix with Q_C.

Constraints

First, we define the modified routing and time-window constraints, which use only binary and integer decision variables. Because the aim is to arrive at an unconstrained formulation, they are not enforced as hard constraints but integrated as penalty terms in the objective/cost function.

Routing constraints

(1) exactly one travelled edge at the first time step:

$$\sum_{v \in \mathcal{C}} x^1_{0,v} = 1,$$

(2) exactly one travelled edge at the time step i:

$$\sum_{(u,v) \in \mathcal{C}^2} x^i_{u,v} = 1 \quad \forall i \in I,$$

(3) exactly one travelled edge at the last time step:

$$\sum_{u \in \mathcal{C}} x^{n+1}_{u,0} = 1,$$

(4) each city left exactly once:

$$\sum_{i \in I} \sum_{v \in \mathcal{C}, v \neq u} x^i_{u,v} + x^{n+1}_{u,0} = 1 \quad \forall u \in \mathcal{C},$$

(5) connecting the edges for the first two time steps:

$$\sum_{(v,w) \in \mathcal{C}^2} x^1_{0,v}\, x^2_{v,w} = 1,$$

(6) connecting the edges for the time steps i and $i+1$:

$$\sum_{(u,v,w)\in C^3} x_{u,v}^{i}\, x_{v,w}^{i+1} = 1 \quad \forall i \in I \setminus \{\mathrm{n}\},$$

(7) connecting the edges for the last two time steps:

$$\sum_{(u,v)\in C^2} x_{u,v}^{n}\, x_{v,0}^{n+1} = 1.$$

Note that the last three sets of constraints imply that each city is not only left exactly once but also entered exactly once. As explained in the previous section, these equality constraints can be transformed into penalty terms. For the quadratic constraints (5) - (7) it should be noted that the constraints (1) - (3) imply "≤", such that the square can be omitted. Overall, one can get a single penalty term for the routing constraints

$$p_R\, x^{\top} Q_R x + p_R\, c_R,$$

where $Q_R \in \mathbb{R}^{s\times s}$ is the quadratic coefficient matrix, $x \in \{0,1\}^s$ the vector of the decision variables, $p_R > 0$ the penalty coefficient and $c_r > 0$ an additive constant.

Time-window constraints

It needs to be ensured that the discrete arrival times $A^i \in \mathbb{N}$ lie within the time windows. In contrast to the MILP formulation, waiting times $\omega^i \in \mathbb{N}$ must be explicitly taken into account. First, notice that the arrival times can be defined recursively, and ultimately be expressed only by the corresponding waiting times and travel costs

$$A^1 = \sum_{v\in C} c_{0v}\, x_{0,v}^{1},$$

$$A^i = \sum_{v\in C} c_{0v}\, x_{0,v}^{1} + \sum_{t=1}^{i-1} \omega^t + \sum_{t=2}^{i} \sum_{(u,v)\in C^2} c_{uv}\, x_{u,v}^{t},$$

for $i \in I$. To enforce the time-window constraints it is enough to require (again for $i \in I$)

$$\sum_{v\in C} e_v\, x_{0,v}^{1} \le A^1 + \omega^1 \quad \text{and} \quad A^1 \le \sum_{v\in C} l_v\, x_{0,v}^{1},$$

$$\sum_{(u,v)\in C^2} e_v\, x_{u,v}^{i} \le A^i + \omega^i \quad \text{and} \quad A^i \le \sum_{(u,v)\in C^2} l_v\, x_{u,v}^{i}.$$

Waiting times can be omitted in the constraints of the latest allowed service time, as there is a trick that involves using the next earliest allowed service time, *Salehi et al. (2022)*. Next, we introduce slack variables $S^{e,i} \in \mathbb{N}$ and $S^{l,i} \in \mathbb{N}$ for $i \in I \cup \{1\}$. By using the representation of A^1 from the above equation one obtains the equality constraints

$$(8) \sum_{v\in C} e_v\, x_{0,v}^{1} + S^{e,1} = \sum_{v\in C} c_{0v}\, x_{0,v}^{1} + \omega^1,$$

$$(9) \sum_{v\in C} l_v\, x_{0,v}^{1} + S^{l,1} = \sum_{v\in C} c_{0v}\, x_{0,v}^{1}.$$

Analogously, let $i \in I$ and use the representation of A^i from the above equation to obtain

$$(10) \sum_{(u,v)\in C^2} e_v\, x_{u,v}^{i} + S^{e,i} = \sum_{v\in C} c_{0v}\, x_{0,v}^{1} + \sum_{t=1}^{i-1} \omega^t + \sum_{t=2}^{i} \sum_{(u,v)\in C^2} c_{uv}\, x_{u,v}^{t} + \omega^i,$$

$$(11) \sum_{(u,v)\in C^2} l_v\, x_{u,v}^{i} + S^{l,i} = \sum_{v\in C} c_{0v}\, x_{0,v}^{1} + \sum_{t=1}^{i-1} \omega^t + \sum_{t=2}^{i} \sum_{(u,v)\in C^2} c_{uv}\, x_{u,v}^{t}.$$

By realizing that the waiting times and the slack variables are bounded by $\max\{ l_u : u \in \mathcal{C}\}$, it becomes clear that they each can be encoded into $\delta := \lfloor \log_2(\max\{ l_u : u \in \mathcal{C}\}) \rfloor + 1$ binary variables. Thus, the above constraints can be written as equality constraints with only binary variables. To sum up, the decision variables $x_{u,v}^i$ from above need to be complemented by the following binary decision variables

$\omega_k^i \in \{0,1\}$	$\forall k \in \{1, \dots, \delta\},\ i \in I \cup \{1\},$	binary representation of waiting time ω^i,
$S_k^{e,i} \in \{0,1\}$	$\forall k \in \{1, \dots, \delta\},\ i \in I \cup \{1\},$	slack variables for earliest-arrival constraints,
$S_k^{l,i} \in \{0,1\}$	$\forall k \in \{1, \dots, \delta\},\ i \in I \cup \{1\},$	slack variables for latest-arrival constraints.

In general, to ensure the compliance of the time-window constraints it is enough to consider the constraints (8) - (11). When formulated with binary variables only, i.e. expressing the slack variables with binaries, they can easily be combined into a single penalty term for the time-window constraints

$$p_T\, x^\top Q_T\, x + p_T\, c_T,$$

where $Q_T \in \mathbb{R}^{d \times d}$ is the quadratic coefficient matrix, $p_T > 0$ the penalty coefficient and $c_T > 0$ an additive constant. The number of binary variables needed is

$$d := s + 3n\delta = n^3 - 2n^2 + 3n + 3n\delta = n\big(n^2 - 2n + 3(1+\delta)\big).$$

i.e., there are s routing variables and $3n\delta$ slack variables being employed to integrate the time-window constraints into the objective.

QUBO Objective

After embedding Q_R and Q_C into $\mathbb{R}^{d \times d}$, one can now formulate the final objective as

$$\min_{\tilde{x} \in \{0,1\}^d} \tilde{x}^T (Q_C + p_R\, Q_R + p_T\, Q_T) \tilde{x}\,.$$

The penalty parameters p_R and p_T must be chosen sufficiently large to enforce feasibility but not so large as to overshadow the original cost term Q_C. In our benchmark study we varied over these parameters to determine the optimal combination. Based on the standard choice $p_R = \max|Q_C| = \max_{u,v} c_{u,v}$, we will refer to the renormalized penalties as

$$P_R := \frac{p_R}{\max|Q_C|} \quad \text{and} \quad P_T := \frac{p_T \cdot \max|Q_T|}{\max|Q_C|}.$$

With this notation, choosing $P_R \approx P_T$ should give both penalties similar weights in the QUBO matrix.

In summary, the edge-based QUBO formulation of the TSPTW requires on the order of $\mathcal{O}(n^3)$ binary variables to represent routing decisions and an additional $\mathcal{O}(n \log(l_{max}))$ variables for encoding the time-window constraints, where $l_{\max} = \max\{ l_u : u \in \mathcal{C}\}$. While this growth limits practical solvability to small instances, both for simulating annealing on classical hardware and for current quantum annealers, the formulation nonetheless establishes a systematic and hardware-compatible representation of the problem. In the following section, the performance of classical solvers and simulated annealing on these QUBOs with realistic problem instances derived from the cargo routing use case is investigated.

5. Analysis and Results

In general, problem instances for the TSPTW must consist of a matrix containing the travel times/costs $c_{u,v}$, as well as two lists, with the earliest (e_u) and the latest (l_u) service time for the ports, which define the time windows. For our study, problem instances were carefully configured in consultation with the Harren Group. They provided us with data on actual shipping routes, including detailed arrival and departure times for various ports as well as estimated realistic time windows. This operational data formed the basis for the creation of our TSPTW test instances. For most of the ports, a connecting route

existed within the operational data, such that the travel time could be derived from them directly. We observed a speed of 10 kn on average. In order to complete the cost matrix with travel times/costs between all ports, the travel time for missing connections between ports was estimated using distances from *sea-distances.org* and the average speed of 10 kn. As a preprocessing step, the resulting cost matrices and time windows were scaled down and rounded to natural numbers to decrease the number of slack variables of the QUBO formulations. The resulting dataset contains 130 problem instances in total, with varying size (n as the number of customers/ports) and scaling. The problem size varied between $n = 2$ and $n = 7$, and for the scaling we used maximal values of $10, 15, 20, 50, 60$ and 70.

For the computational SA results we used the QUBO formulation from the previous section 4 on a Dell Precision 7865 Tower with an AMD Ryzen Threadripper PRO 5965WX CPU with 24-Cores @ 3.8 to 4.5 GHz, 128GB RAM and a NVIDIA RTX A6000.

The classical CPLEX results were obtained by solving the MILP formulation of the previous section 4 with CPLEX 12.10.0.0 (standard settings) on a Dell Inc. Latitude 5411 Laptop with Intel(R) Core(TM) i7-10850H CPU @ 2.70GHz, 16GB RAM and a NVIDIA GeForce MX250. The difference in hardware used for each computation method devalues a comparison of absolute performance metrics, however, the scaling of such metrics with problem size should still be comparable.

Fig.1 shows the density of the QUBO matrices (ignoring the diagonal entries) for all problem instances, separated into contributions from the various components of the objective: routing constraints, time-window constraints, and the combined formulation. For each instance, the three overlapping bars illustrate how the density evolves when the components are included. While the routing part alone remains comparatively sparse, the time-window component is consistently very dense, with values ranging between 0.45 and 0.8 across all instances. Consequently, the full QUBO inherits this high density, indicating that the encoding of time-window feasibility dominates the structural complexity of the problem.

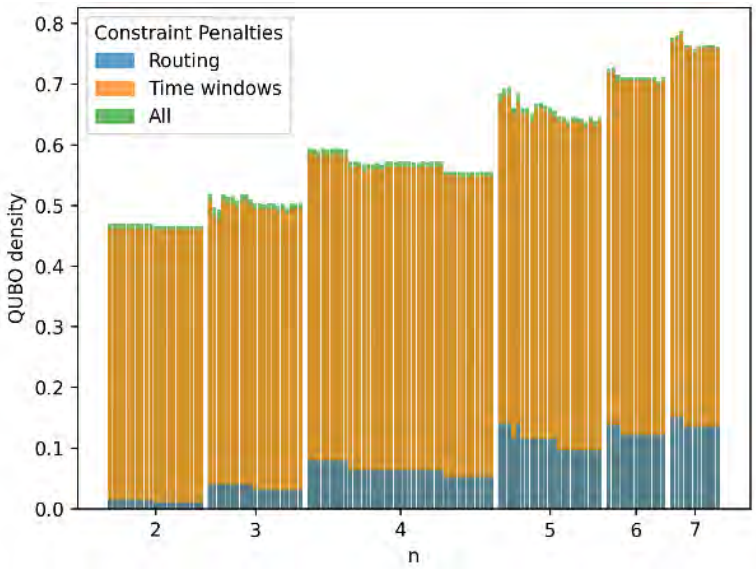


Fig.1: Density of the QUBOs itemized by the various components of the objective

To analyze the effect of penalty parameters on solution quality, Figure 2 shows heatmaps of likelihood of feasibility, averaged over all instances of $n = 4$, where on the axes the normalized penalty parameters are varied. Three perspectives are presented: (a) Routing feasibility: A clear dependence on the routing penalty is observed. High values of P_R combined with small values of P_T yield the best routing feasibility, indicating that strong penalties are necessary to enforce valid tours, while excessive emphasis on time-window penalties can distort routing behaviour. (b) Time-window feasibility: The pattern is not completely reversed, as the feasibility is best when P_T is equal to 1 and P_R remains small, but the time-window feasibility is in general much lower than the routing feasibility (c) Total feasibility including optimality: Achieving both routing and time-window feasibility, together with finding the

optimal solution, requires a balanced but strong penalization of both constraint classes. The combined heatmap shows that when P_R and P_T are simultaneously large, the optimal solution is found with the highest probability. These results highlight the strong co-dependency of penalty parameters in QUBO formulations. Unlike MILP, where feasibility is enforced directly via constraints, QUBO formulations require careful tuning of penalty magnitudes to balance competing conditions. Too strong a bias toward one constraint class undermines the other, while too weak penalties overall reduce the probability of finding feasible ground states. The observed trade-offs emphasize the challenge of penalty selection, a well-known bottleneck in QUBO modelling.

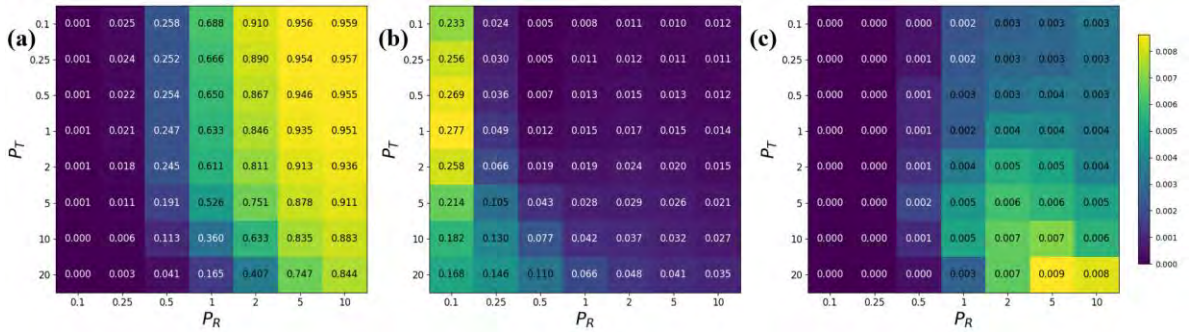


Fig.2: Heatmaps showing the average probability that a sample for a problem instance of size $\boldsymbol{n = 4}$ satisfies (a) the routing constraints, (b) the time-window constraints, and (c) satisfies all constraints and is the optimal solution.

The Time-to-Solution (TTS) metric quantifies the expected wall-clock time until the solver finds an optimal solution with high confidence, *Albash and Lidar (2018)*. Suppose a single run of a probabilistic algorithm (e.g. simulated annealing or quantum annealing) finds an optimal solution with probability $p \in (0,1)$. To achieve a confidence level α (e.g. $\alpha = 0.999$), the expected number of independent runs R can be calculated by

$$(1-p)^R = (1-\alpha) \quad \Rightarrow \quad R = \frac{\ln(1-\alpha)}{\ln(1-p)}.$$

If one run of the algorithm requires time t, the TTS is then defined as

$$\mathrm{TTS}(\alpha) := t \cdot R = t \cdot \frac{\ln(1-\alpha)}{\ln(1-p)}.$$

This metric accounts both for the stochastic nature of heuristic solvers and the computational cost of individual runs. Fig.3 presents the Time-to-Solution (TTS) for Simulated Annealing (SA) and CPLEX as a function of the problem size n. The violin plots show the distribution of TTS values across all generated instances, while dashed regression lines highlight the scaling trend. For the scaling trend an exponential fit was calculated, that is a function ab^n which minimizes the square errors. Hence, the slope in the legend of the plot equates the base b. The annotated parameter tuples represent the best-performing normalized penalty parameters selected per problem size, which were used for the SA results in the plot.

For SA, results could only be obtained up to $n = 5$, as for larger instances no feasible solutions were found by that method. The TTS increases steeply with problem size, reflected by the regression slope of 2.379 in logarithmic scale, indicating exponential growth in runtime even for small problem instances. Moreover, the spread of the violins widens with n, suggesting that some problem instances are significantly harder to solve than others. In contrast, CPLEX demonstrates consistently low TTS values across all tested problem sizes up to $n = 7$. The regression slope of 0.050 is nearly flat, confirming that state-of-the-art MILP solvers are highly efficient for small-sized TSPTW instances. This aligns with previous work showing that exact MILP methods are competitive up to a few hundred nodes before scaling issues become prohibitive, *Hungerländer and Truden (2018).*

The contrasting trends highlight a central limitation of the QUBO approach: while CPLEX handles the linearized TSPTW efficiently for small n, the QUBO grows as $\mathcal{O}(n^3)$ in binary variables, leading to dense couplings and unfavourable scaling for SA. This behaviour has been similarly observed in recent studies on QUBO formulations of routing problems *Salehi et al. (2022), Papalitsas et al. (2019).* According to our assessment the bottleneck lies not primarily in the annealing heuristic itself, but in the QUBO representation of the problem. Furthermore, the logarithmic encoding of the waiting times in binary variables could act as a hurdle for the SA algorithm, where bits are flipped one at a time, resulting in large jumps in the number representations.

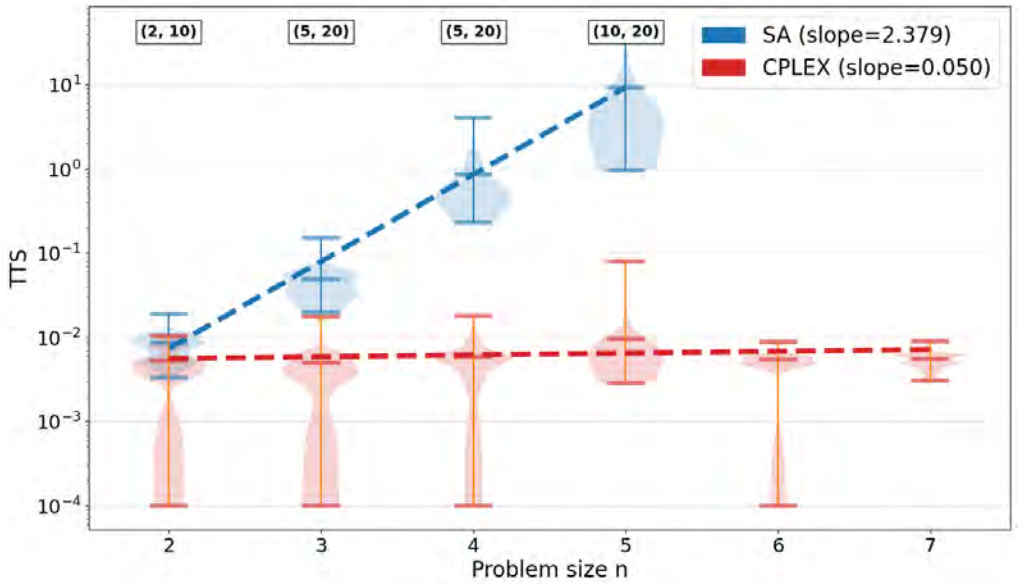


Fig.33: Time-to-Solution (TTS) with $\boldsymbol{\alpha = 0.999}$ for different problem sizes $\boldsymbol{n}$ (number of ports/ customers). For each $\boldsymbol{n}$, the best normalized penalty parameters $(\boldsymbol{P_R}, \boldsymbol{P_T})$ are used for SA.

6. Conclusion and Outlook

In this work we investigated the cargo routing problem for bulk carriers, where fuel consumption depends nonlinearly on vessel speed and routing must satisfy strict time windows. The primary aim was to demonstrate how such problems from maritime practice can be modelled and approached with quantum technology, both to show the way forward for practical adoption and to investigate its current potential. By discretizing vessel speed, the problem was reformulated into a Travelling Salesman Problem with Time Windows (TSPTW) and expressed both as a MILP and as a QUBO, enabling a direct comparison between classical optimization, quantum-inspired and quantum approaches.

For small but representative problem instances, our results show that the Time-to-Solution (TTS) of simulated annealing (SA) scales poorly for the QUBO formulations of the cargo routing problem. This behaviour is presumably linked to the dense structure of the QUBO and the logarithmic encoding of integers used, which generates rugged energy landscapes and reduces the probability of reaching ground states efficiently. The time-window constraints in particular are responsible for the high density, introducing an all-to-all connectivity between the involved slack variables. Similar challenges have been observed in prior work, e.g. by *Salehi et al. (2022),* who report that dense couplings, long chain lengths, and the difficulty of selecting appropriate penalty parameters strongly affect solution quality for TSPTW formulations on both simulated and quantum annealers.

These findings indicate that while unconstrained binary formulations provide a systematic and hardware-compatible representation of routing problems, they remain challenging to solve efficiently with current classical heuristics and quantum annealers. Promising directions for future research include problem-specific encodings, improved penalty strategies, and hybrid optimization methods. In particular, decomposition techniques such as Benders decomposition, *Benders (1962),* could be leveraged to

partition the problem into more tractable subproblems, with time-window constraints potentially delegated to specialized subroutines. Combining such strategies with advancing quantum hardware may enable quantum optimization to complement established exact and heuristic approaches for large-scale maritime routing. This work represents a first step in that direction, bringing maritime logistics closer to benefiting from quantum-enabled optimization.

Acknowledgments

This work has been supported by the programme "Quantum computing for shipping and maritime logistics in Hamburg", which was funded by the Free and Hanseatic City of Hamburg. Special thanks go to the shipping company Harren Group and its managing director Nils Aden, who provided us with data and crucial background information on the use case. Furthermore, we like to thank Valeria Bartsch for many useful comments and Hai-Yen Van and Jachin Tekle Gemta for their support in preparing data and plots.

References

ALBASH, T.; LIDAR, D. A. (2018), *Demonstration of a Scaling Advantage for a Quantum Annealer over Simulated Annealing,* Phys. Rev. X 8/3, p. 031016

BELOTTI, P.; LEE, J.; LIBERTI, L.; MARGOT, F.; WÄCHTER, A (2009), *Branching and bounds tightening techniques for non-convex MINLP*, Optimization Methods and Software 24/4–5, pp.597–634

BENDERS, J. F. (1962), *Partitioning procedures for solving mixed-variables programming problems,* Numerische Mathematik 4, pp. 238-252

CHRISTIANSEN, M.; FAGERHOLT, K.; PISINGER, D. (2025), *Fifty Years on Maritime Transportation*, EURO J. Transportation and Logistics 14, p. 100148

DAM, W.; MOSCA, M.; VAZIRANI, U. (2001), *How Powerful is Adiabatic Quantum Computation?,* 42nd IEEE Symp. Foundations of Computer Science, pp.279-287

DENCHEV, V. S.; BOIXO, S.; ISAKOV, S. V.; DING, N.; BABBUSH, R.; SMELYANSKIY, V.; MARTINIS, J.; NEVEN, H. (2016), *What is the Computational Value of Finite-Range Tunneling*?, Phys. Rev. X 6/3, p. 031015

DESROSIERS, J.; DUMAS, Y.; SOLOMON, M. M.; SOUMIS, F. (1995), *Chapter 2 Time Constrained Routing and Scheduling*, Handbooks in Operations Research & Management Science 8, Elsevier, pp.35-139

DUMAS, Y.; DESROSIERS, J.; GÉLINAS, E.; SOLOMON, M. M. (1995), *An Optimal Algorithm for the Traveling Salesman Problem with Time Windows*, INFORMS: Operations Research 43/2, pp.367-371

FELD, S.; ROCH, C.; GABOR, T.; LÜCK, T.; SEIDEL, C.; NEUKART, F; GALTER, I; MAUERER, W; LINNHOFF-POPIEN, C. (2019), *A Hybrid Solution Method for the Capacitated Vehicle Routing Problem Using a Quantum Annealer*, Frontiers in ICT 6

GAMBARDELLA, L. M.; TAILLARD, É.; AGAZZI, G. (1999), *MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows,* https://people.idsia.ch/~luca/tr-idsia-06-99.pdf

HAYASAKA, H.; IMOTO, T.; MATSUZAKI, Y.; KAWABATA, S. (2024), *Quantum Annealing Showing an Exponentially Small Success Probability Despite a Constant Energy Gap with Polynomial*

Energy, Phys. Rev. A 110/2, p. 022620

HO, S.C..; HAUGLAND, D. (2004), *A Tabu Search Heuristic for the Vehicle Routing Problem with Time Windows and split deliveries*, Computers & Operations Research 31/12, pp.1947-1964

HUNGERLÄNDER, P.; TRUDEN, C. (2018), *Efficient and Easy-to-Implement Mixed-Integer Linear Programs for the Traveling Salesperson Problem with Time Windows*, Transportation Research Procedia 30, pp.157-166

KARA, I; DERYA, T (2015), *Formulations for Minimizing Tour Duration of the Traveling Salesman Problem with Time Windows,* Procedia Economics and Finance 26, pp.1026-1034

KIRKPATRICK, S.; GELATT, C.D.; VECCHI, M.P. (1983), *Optimization by Simulated Annealing*, Science 220/4598, pp.671-680

LUCAS, A. (2014), *Ising Formulations of Many NP Problems*, Frontiers in Physics 2/5

MASUDA, K.; TSUYUMINE, Y.; KITADA, T.; HACHIKAWA, T.; HAGA, T (2023), *Optimization of delivery plan by quantum computing,* Optimization 85/1

McGEOCH, C.C. (2014), *Adiabatic Quantum Computation and Quantum Annealing: Theory and Practice*, Springer Nature

NEWMAN, J.N. (2018), *Marine hydrodynamics,* MIT Press

NEUKART, F.; COMPOSTELLA, G.; SEIDEL, C.; VON DOLLEN, D.; YARKONI, S.; PARNEY, B. (2017), *Traffic Flow Optimization Using a Quantum Annealer*, Frontiers in ICT 4

OHLMANN, J.W.; THOMAS, B.W. (2007), *A Compressed-Annealing Heuristic for the Traveling Salesman Problem with Time Windows*, INFORMS J. Computing 19, pp.80-90

OSMAN, I. H. (1993), *Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problem*, Annals of Operations Research 41, pp.421-451

PAPALITSAS, C.; ANDRONIKOS, T.; GIANNAKIS, K.; THEOCHAROPOULOU, G.; FANARIOTI, S. (2019), *A QUBO Model for the Traveling Salesman Problem with Time Windows*, Algorithms 12/10, p. 224

SALEHI, Ö.; GLOS, A.; Miszczak, J. A. (2022), *Unconstrained Binary Models of the Travelling Salesman Problem Variants for Quantum Optimization*, Quantum Information Processing, 21/2, p. 67

SCHNEEKLUTH, H.; BERTRAM, V. (1998), *Ship Design for Efficiency and Economy*, Butterworth-Heinemann

SIRIMANNE, S.N.; HOFFMAN, J.; JUAN, W; ASARIOTIS, R; ASSAF, M; AYALA, G; AYOUB, A; BENAMARA, H; CHANTREL, D; HOFFMANN, J; et al. (2020), *Review of Maritime Transport 2020*, United Nations Conf. on Trade and Development (UNCTAD)

SZAL, O.; RUBBERT, S.; RIZVANOLLI A. (2025), *Benchmarking the Maritime Inventory Routing Problem on a Quantum Annealing-Hybrid System,* TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation 19/1, pp. 113-122

VENTURELLI, D.; MARCHAND, D. J. J.; ROJO, G. (2015), *Quantum Annealing Implementation of Job-Shop Scheduling*, arXiv preprint arXiv:1506.08479

YOUNG, C; ZHENG, Y; YEH, C.; JANG, S. (2007), *Information-Guided Genetic Algorithm Approach to the Solution of MINLP Problems,* Ind. Eng. Chem. Res. 46/5, American Chemical Society

ZHANG, Y.; SAHINIDIS, N.V. (2024), *Solving continuous and discrete nonlinear programs with BARON,* Computational Optimization and Applications, Springer, pp.1-39

Early-Stage Productivity Evaluation through Model-Based Shipbuilding: Connecting NAPA Models with Production Simulation

Kohei Matsuo, National Maritime Research Institute, Tokyo/Japan, kohei@m.mpat.go.jp
Tomoyuki Taniguchi, National Maritime Research Institute, Tokyo/Japan, taniguchi-t@m.mpat.go.jp
Mizuki Morishita, National Maritime Research Institute, Tokyo/Japan, morishita-m@m.mpat.go.jp
Takayoshi Masui, NAPA Ltd., Helsinki/Finland, takayoshi.masui@napa.fi
Kazunori Aoki, NAPA Japan Ltd., Kobe/Japan, kazunori.aoki@napa.fi

Abstract

This study introduces a model-based approach to shipbuilding, focusing on its application at the early production planning stage through simulation. The key idea is to shift from conventional visual- and experience-based planning, relying on 2D drawings or 3D visualization tools, to a model- and simulation-based process that uses early 3D product models enriched with process information. This enables quantitative and rational assessment of productivity and manufacturability from the very first ship. A case study on an engine-room double-bottom block demonstrates the workflow and highlights both numerical differences in assembly strategies and qualitative insights for design and production. The approach provides a foundation for achieving efficiency, quality, and safety from the outset of ship construction.

1. Introduction

The development of future ships is expected to become increasingly complex, encompassing energy-efficient vessels, alternative-fuel ships, and autonomous ships, while simultaneously meeting society's demand for rapid customization to individual customer requirements. As a result, shipyards are now under mounting pressure to shorten construction lead times, improve productivity with limited human resources, and respond flexibly to diverse customer needs.

In such an environment, it becomes essential to estimate production productivity accurately and rationally even at the earlier stages of product development. Maximizing efficiency from the very first ship by optimizing production methods contributes directly to competitiveness. The fundamental approach to achieve this is to base production planning and simulation directly on the as-built 3D product models, thereby enabling precise and detailed analyses.

However, the current state of shipyards reveals persistent challenges. As illustrated in Figure 1, which depicts the typical trend of man-hours across series ships, the actual cost of the first ship is not only high but often significantly exceeds the initial estimates. This occurs because man-hours and costs are typically derived from past data and rough quantity indicators (such as block weight or welding length) rather than from precise product-specific information that reflects the actual production procedure. Such estimation methods rapidly lose validity as the specifications diverge from previous or similar vessels. Moreover, the inherent lack of experience in constructing brand-new ship exacerbates the situation, leading to unanticipated increases in man-hours and costs. In summary, shipyards face structural problems: insufficient data for accurate estimation (detailed data from production design systems often arrive too late, close to construction), design errors that necessitate rework, process disruptions caused by inadequate quality control, and an excessive reliance on workers' adaptability in the face of unclear or incomplete instructions.

Shipbuilding is a quintessential example of Engineering-to-Order (ETO) manufacturing. Unlike industries where prototypes or pilot runs are feasible, shipyards must proceed directly to full-scale production without the benefit of trial manufacturing. While this is a defining feature of shipbuilding, it does not imply that inefficiencies in the first ship are inevitable. On the contrary, the goal should be to achieve maximum efficiency from the very first vessel, on par with or even exceeding that of later, more familiar series ships. This represents a fundamental proposal for advancing manufacturing in shipbuilding.

To address these challenges and to ensure that cost, delivery, quality, and safety are achieved as planned, it is crucial to evaluate manufacturability and productivity in the earlier design stage. This requires a shift toward model-based approach, in which production planning is grounded in digital product models prepared at an earlier stage. By enabling thorough meticulous advance planning, shipyards can move away from ad hoc construction and instead aim to produce the first ship in comparable man-hours and costs to subsequent vessels in the series. This aligns with the widely recognized principle that early-stage decisions exert a disproportionate influence on the overall success of product development.

Realizing this approach may involve or be supported by several enabling technologies: early-stage digital modeling, shipbuilding-specific Bill of Materials (BOM)/ Bill of Process (BOP) systems, production simulation tools, Product Lifecycle Management (PLM)-based integration of design and manufacturing data, and digital quality management. Among these, the present study focuses particularly on early modeling techniques, BOM/BOP methodologies, and production simulation technologies. A demonstrative case study is presented using an engine-room double-bottom block model, highlighting how early production planning can be reformulated based on 3D product models and associated digitalized process data (Engineering BOM [E-BOM] to Manufacturing BOM [M-BOM], and BOP), and quantitatively evaluated through simulation to pursue maximum efficiency from the first ship.

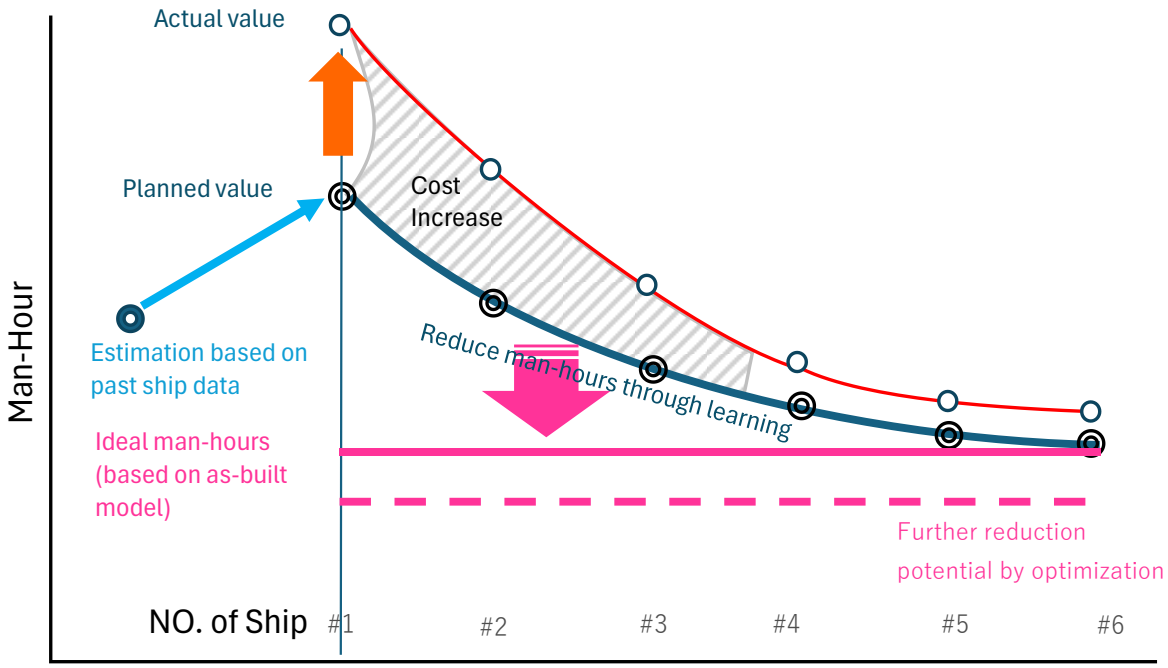


Fig.1: Man-hour transition in series ship construction and target state of this study

2. Model-Based Approach for Production Planning

This chapter outlines the concept of a model-based approach to production planning as proposed in this study and introduces the fundamental technologies that can support its realization. Particular emphasis is placed on the production simulation technologies currently being developed at National Maritime Research Institute (NMRI), which serve as a key enabler for the practical implementation of this approach.

2.1. Model-Based Approach

In this study, the concept and definition of a model-based approach for ship production planning are clarified. The term is used in the same sense as commonly employed expressions such as Model-Based Systems Engineering (MBSE), Model-Based Design (MBD), or the emerging Model-Based Approval

(MBA). In all these, the central idea is to move away from relying solely on 2D drawings and instead ground engineering and business processes in 3D digital models enriched with associated information.

The fundamental principle of the model-based approach as applied in this research can be summarized as follows:

- Basing analyses on the 3D product model of the vessel under design.
- Instead of relying on models of past or sister ships, this approach requires the preparation of a digital model that faithfully incorporates the specifications of the vessel to be built. Productivity and manufacturability are then examined directly with reference to this model.
- Conducting detailed production simulations grounded in the model.
- Rather than relying on rough estimations derived from past data and aggregated indicators such as total weight or welding length, the model-based approach emphasizes specific and concrete analysis. The digital model is used to "envision the actual production process" in detail, reconstructing individual tasks and accumulating them to produce rational estimates of man-hours and costs.

More concretely, the model-based approach for ship production planning entails making effective use of the 3D product model that is gradually established during design. This model is employed in the computational environment to reproduce the production activities that will take place in the shipyard. Achieving this requires both the early availability of adequately detailed 3D models and simulation technologies capable of reproducing shipyard manufacturing processes with high fidelity.

2.2. Production Simulation

At National Maritime Research Institute (NMRI), advanced shipbuilding production simulation technologies have been developed to reproduce shipyard operations with high fidelity. These technologies enable the creation of a virtual factory environment in which shipbuilding processes—including not only main tasks but also ancillary tasks such as preparation and setup—are simulated in detail. The simulation framework is based on multi-agent technologies, which allow for the representation of the complex interactions among workers, equipment, and processes.

A distinguishing feature of shipbuilding is its low ratio of main tasks to total tasks. A significant portion of labor is consumed by ancillary tasks, which depend heavily on workers' situational judgments and are not directly correlated with physical quantities of the product. In contrast to many other industries where automation and robotics cover a large share of operations, shipbuilding involves a wide range of complex manual work. While main tasks scale with material-based quantities such as block weight or welding length, ancillary tasks vary substantially depending on the design, layout, and working conditions of each vessel—even when the overall material-based quantities are similar. Consequently, a production simulation for shipbuilding must accurately represent ancillary tasks alongside main tasks, ensuring that the model reflects the real complexity of the shipyard environment.

The production simulator itself is designed as a general-purpose computational tool for reproducing ship production within a virtual factory. It functions as a fundamental solver of shipbuilding operations. Importantly, the simulator is not tied to a single application: how it is deployed in shipyard practice depends on user requirements and scenarios. Potential use cases span the entire shipbuilding lifecycle, from upstream design stages to downstream shop-floor operations.

Representative applications include the following:

- Early production planning:
  In parallel with functional design, block division, assembly sequences, construction methods, and resource allocation can be tested virtually on the block models of the vessel under design, allowing multiple alternatives to be explored computationally in advance. The outcome can

be fed back into design, and also used for planning assembly locations and balancing the overall schedule.

- Detailed production planning & preparation:
  In parallel with production design, simulations can be performed using the finalized 3D production models of each block or compartment. These simulations derive theoretical man-hours and schedules based on modeled processes and allow the results of earlier production planning to be verified and refined. They support scheduling, evaluation of more efficient production measures, and preparation of work instructions, as well as more detailed planning for scaffolding, jigs, block lifting, and block supports.
- On-site work management:
  Simulation outputs can be shared with production workers to familiarize them with actual construction methods in advance, thereby serving as a tool for training, skill transfer, and work preparation.
- Factory and equipment planning:
  When introducing new factories, facilities, or equipment—or when reconfiguring layouts—the simulator enables virtual re-arrangements and evaluation of their effects. The investment's effectiveness and payback period can then be quantitatively assessed before implementation.

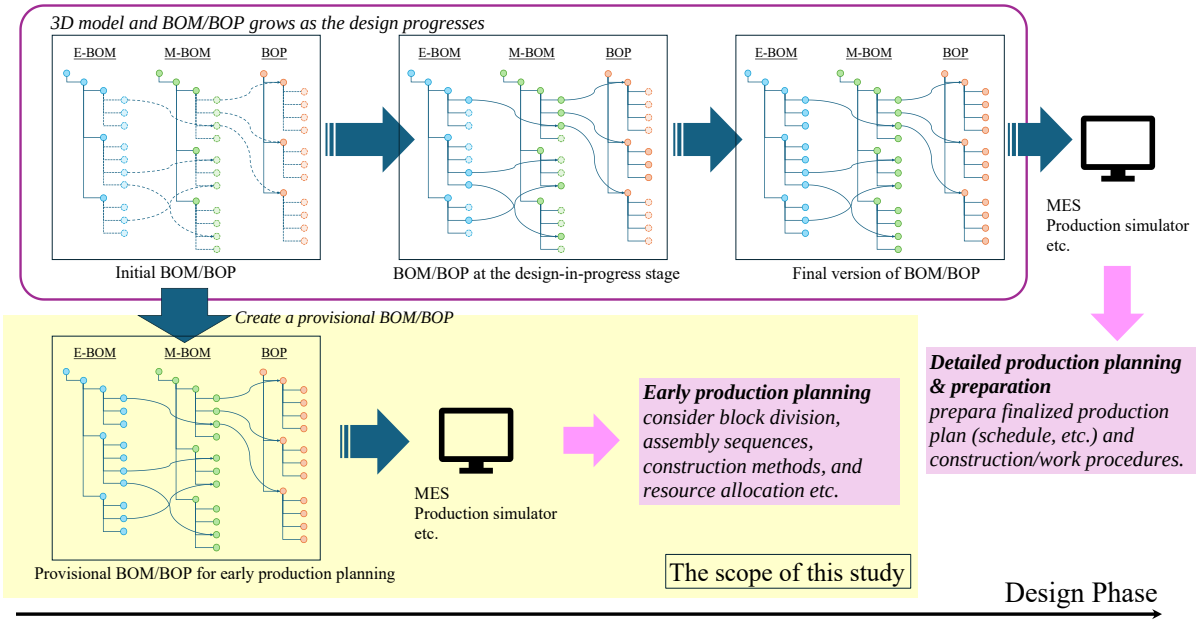


Fig.2: Conceptual use cases of production simulation across processes

In sum, NMRI's production simulation technologies provide a versatile and foundational platform for analyzing, planning, and optimizing shipbuilding processes at multiple levels, from strategic decision-making to worker training.

2.3. Simulation for Early Production Planning

This study focuses in particular on the application of production simulation during the early production planning stage of ship construction. At this stage, which coincides with the functional design phase of the ship (e.g., specification review, general arrangement (GA), machinery arrangement (MA) studies, and structural class approval), the fundamental construction policy of the ship is determined. This involves block division, defining assembly strategies and sequences for each block, and assigning them to appropriate production lines and stages. Based on these decisions, resource loads across factories, and construction stages are assessed, a preliminary factory schedules are created, and corresponding budget plans for the production department are formulated.

The central challenge addressed here is the maximization of efficiency from the very first ship. Although it is well known that the majority of a ship's total cost is determined at this early stage, conventional production planning at this point has been hampered by the low accuracy of available information and reliance on qualitative evaluations. This often makes optimization difficult and allows critical issues to go unnoticed until later stages. In current practice, experienced engineers typically develop plans by inspecting 2D drawings or relying on 2D/3D visualization tools, with planning outcomes heavily dependent on personal expertise and intuition (experience-based planning). Once block divisions or assembly sequences are set, they are seldom changed in later ships due to the excessive rework involved. In contrast, this study proposes to transform production planning from experience-driven approaches toward a simulation-based, quantitative methodology. By using production simulation technologies, it becomes possible to test multiple alternatives computationally, evaluate their impacts in measurable terms, and make rational, evidence-based decisions at the earlier design stages. This transition from As-Is (visual- and experience-based planning) to To-Be (simulation-based planning) represents the key gap to be bridged, ultimately supporting the optimization of Quality, Cost, Delivery, and Safety (QCDS) from the outset.

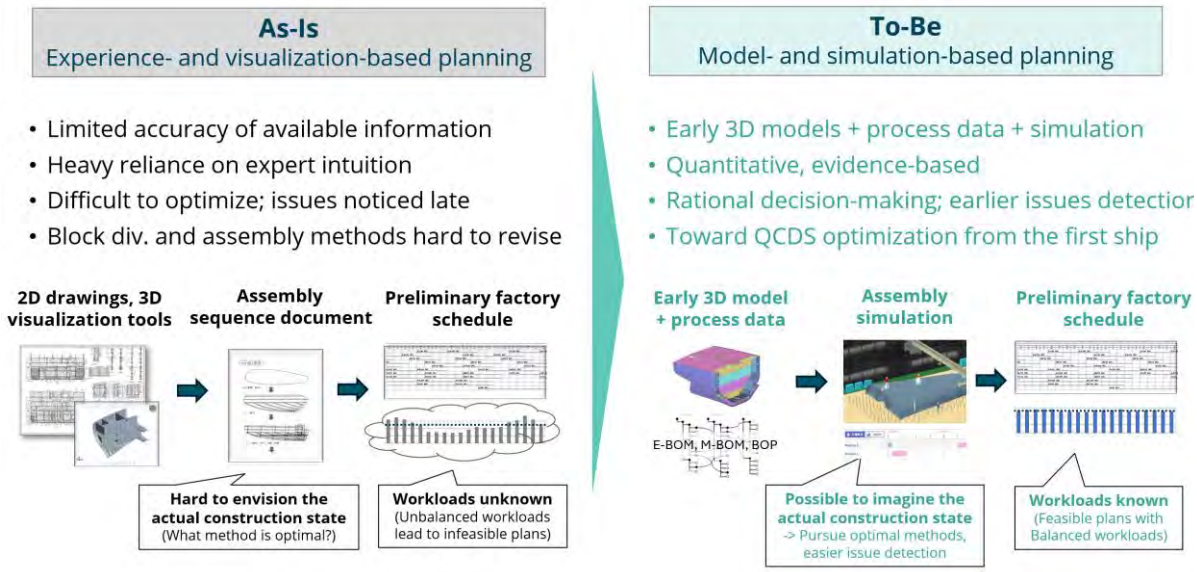


Fig.3: Purpose of simulation for early production planning (from As-Is to To-Be)

3. System Implementation of Model-Based Approach

This chapter discusses the system implementation of the model-based approach for ship production planning introduced in Chapter 2. Specifically, it focuses on establishing an implementation environment for production simulation at early production planning and presents a concrete case study of productivity evaluation based on digital models. Through this demonstration, the practical feasibility of applying model-based approach to early-stage production planning is illustrated.

3.1. Data Preparation and Level of Detail (LOD)

Before implementing the model-based approach, it is essential to clarify the prerequisites: what types of information and to what degree of model detail are required to enable production simulation as described in this study. As outlined in Chapter 2, the model-based approach relies on simulating shipyard operations within a virtual factory environment. To do so, three categories of fundamental information are needed: product information, process information, and resource information.

In general terms, product information corresponds to E-BOM, process information to the M-BOM andBOP, and resource information to the Bill of Equipment (BOE). Each shipyard progressively refines and finalizes these elements during the design and production planning phases. The key question

addressed here is: what level of BOM/BOP detail must users prepare in order to realize simulation at the early production planning stage? Defining this requirement corresponds to establishing the Level of Detail (LOD). Importantly, it is not necessary to prepare a complete or finalized BOM/BOP at this stage; rather, an appropriate level of abstraction should be selected depending on the purpose of the analysis.

The purpose, as emphasized in Section 2.3, is to maximize efficiency from the first ship by developing a production strategy at the early planning stage. This entails block division, assembly sequencing, selection of optimal production methods, identification of required resources, creation of preliminary factory schedules, and confirmation of delivery dates for major materials and equipment. As long as these core aspects of ship construction can be visualized and validated, detailed consideration of minor outfitting tasks is unnecessary at this stage. Thus, a coarse but consistent BOM/BOP is sufficient.

For product information (E-BOM), it is sufficient to include major structural members such as plates, stiffeners, and brackets. In hull construction, welding constitutes a major share of work; thus, these data are essential for estimating welding-related workloads. Minor reinforcements may be omitted, although highly numerous elements such as collar plates should be included. Approximate shapes and dimensions are adequate; exact geometries are not required. Plate thickness information, however, is valuable for estimating weld sizes and related workloads. For process information (M-BOM and BOP), the main requirement is to support evaluation of block assembly strategies and reliable estimation of man-hours and schedules. A simplified BOP structure consisting of material preparation => fitting => welding for each product is generally sufficient.

This LOD is closely aligned with the requirements of 3D models prepared for Model-Based Approval (MBA). In practice, such models could also be applied to early production planning simulations. Moreover, when the purpose is only to compare relative efficiency of alternative building methods, even coarser models such as 3D models for global Finite Element Analysis developed in early functional design, can provide sufficient input.

In this study, product information (E-BOM) is assumed to be prepared primarily using 3D CAD systems. As an example, NAPA Steel is employed as a structural design tool. However, other 3D models can be equally applied. For process (M-BOM, BOP) and resource information (BOE), dedicated editor tools have been developed to allow users to generate and manage this data efficiently, as described in the following sections.

Design Stage	Purpose		Required Model Detail (LOD)
	Design	Early Production Simulation	
Concept/Contract Design [Low LOD]	• Ship performance and lightweight estimation • Initial general arrangement and schematic drawings		• Rough model of hull form, compartments and primary members
Functional Design — Early [Medium LOD]	• Main structural arrangements • Rule calculations, global FEM • Block weight estimates • Major outfitting arrangements	• Block division • Preliminary assembly strategy	• Plates, stiffeners, key brackets and openings (with thickness data)
Functional Design — Late / 3D MBA level [High LOD]	• Refined structure (final shape, holes, end-connections, reinforcements) • Approval drawings (**3D MBA**) • Local FEM • Detailed machinery/piping • Downstream 3D CAD linkage	• Assembly sequence • Process planning (Man-hour estimates and preliminary factory schedule) **=> Primary target for early production simulation**	• Medium-LOD + additional details (e.g., slots, end-cuts) • + Simplified BOP (place materials → fitting → weld → turnover, etc) • Enables accurate estimation of weld length (exact geometry not mandatory)

Fig.4: Example of LOD by Design Stages for Early Production Simulation

.

3.2. Connecting NAPA Model with NMRI Production Simulation

The experimental implementation workflow proceeds from importing product data from CAD to generate E-BOM, followed by the creation of M-BOM andBOP, and finally to the execution of the production simulation. If M-BOM and BOP data are already available, they can be directly imported; otherwise, they are generated automatically from the E-BOM.

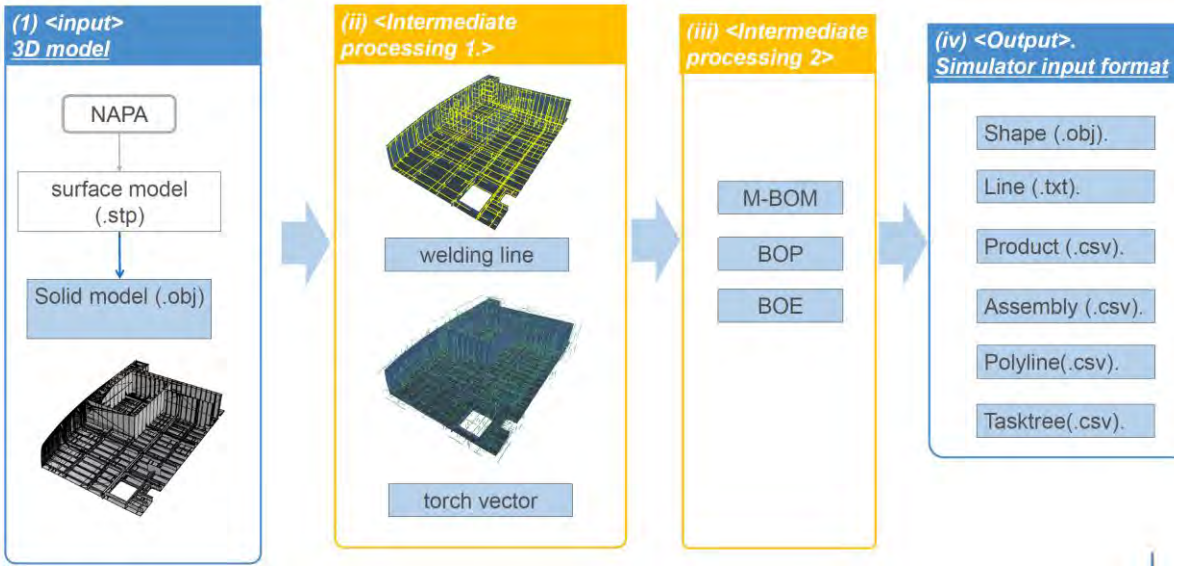


Fig.5: Conceptual data flow leading to production simulation

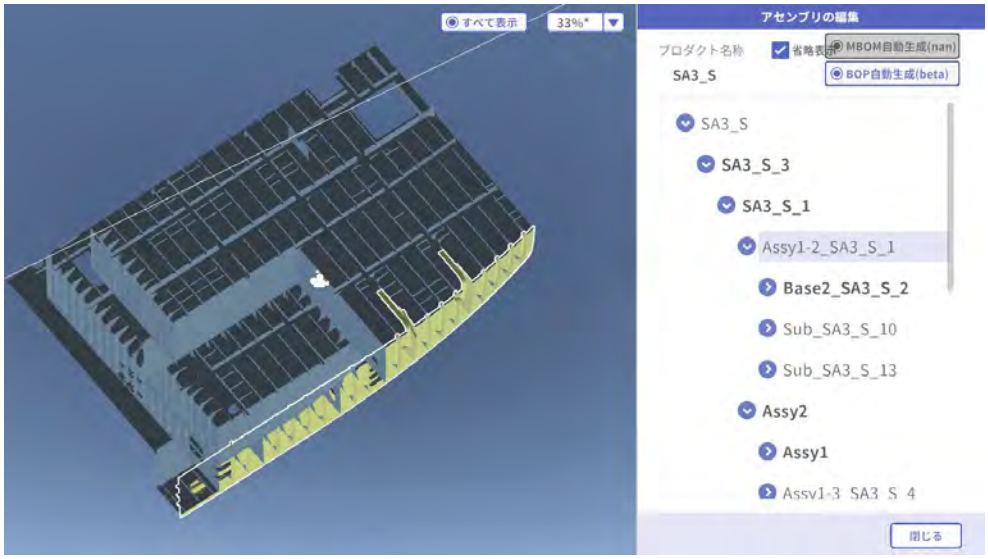
3.2.1. Importing Product Models from NAPA Systems

In this project, we experimentally established a data flow that connects a NAPA Steel model to the NMRI production simulation system, and confirmed the required data processing procedures. The workflow is summarized as follows. First, the 3D geometric model from NAPA Steel (originally a surface model) is exported in STEP (STP) format and converted into a solid model using NMRI's in-house processing system. Next, intersections between structural members are computed on the solid model to define welding lines. This enables the generation of weld-line data even if such definitions are absent in the original design model. For fillet welds, welding lines are defined on both sides of the plate thickness. When plate thickness information is available in the model, it is further used to assign leg length values to the welds.

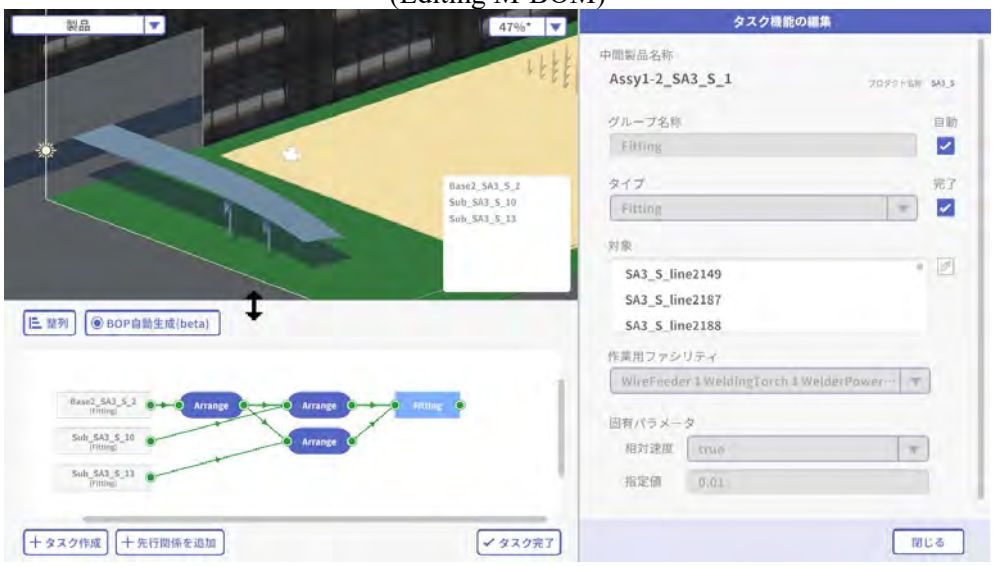
3.2.2. Generation of M-BOM and BOP Data

The corresponding M-BOM and BOP are generated for the specific ship block. These datasets are then converted into a dedicated file format suitable for input into the production simulator.

To support this process, we have developed a BOM/BOP editor that allows users to create M-BOM and BOP data through a visual interface. In addition, an automatic data generation system has been implemented to produce initial drafts of M-BOM and BOP data for typical or similar blocks, thereby reducing the manual workload and accelerating the preparation of production information.



(Editing M-BOM)



(Editing BOP)

Fig.6: Image of the BOM/BOP editor

4. Results – Case Study of Assembly Method Comparison

To demonstrate the feasibility of the model-based approach at the early production planning stage, we conducted a case study that illustrates both the operational image of the system and the potential effects and challenges associated with its application. The focus of this demonstration is on early production planning, particularly the evaluation of assembly methods.

In current shipyard practice, after determining block divisions, planners must define assembly strategies, sequences, and the overall construction order. Based on these decisions, resources such as production lines, stages, and equipment are allocated, and schedules are established to form the foundation of the master assembly plan. Traditionally, this process is carried out by experienced planners using 2D drawings or, more recently, 3D visualization tools. In both cases, planning outcomes rely heavily on personal expertise.

In contrast, our demonstration envisioned performing these tasks through a model- and simulation-based process. The scenario assumed a situation in which a shipyard determines the fundamental construction policy and strategy for a new vessel. The case study targeted the double-bottom block of the engine room of a medium-sized cargo vessel and examined alternative assembly methods.

Two distinct assembly patterns are considered:

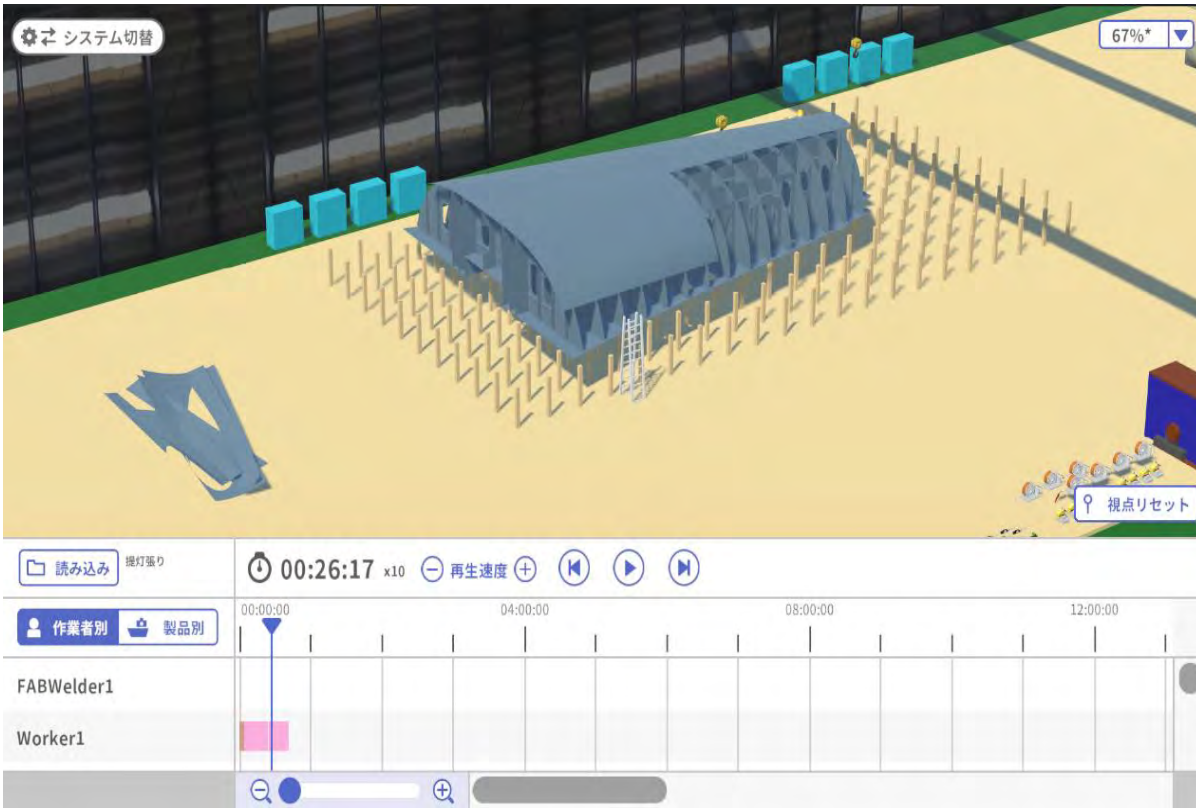
- Pattern 1 (assembly => turnover): The assembly is first carried out with the inverted inner-structure block. The shell plates are then placed and butt-welded onto the framework. Afterwards, the block is turned over, and the remaining welds are executed in the flat (downhand) position.
- Pattern 2 (sub-assembly => assembly): In this method, the shell plates are preassembled and welded together on a separate base to form a shell-plate sub-block, which may be fabricated either prior to or in parallel with the inner-structure block. At the assembly stage, the upright inner-structure block is mounted onto the shell-plate sub-block, and the connection welds between the inner structures and the shell plates are executed.

For the comparison, both assembly patterns are virtually constructed in a 3D environment. The product model used in this demonstration include all structural members and is imported from NAPA Steel. While a fully detailed model is applied here, it is considered that a coarser model could also be sufficient for early production planning. The M-BOM and BOP are created using the BOM/BOP editor. The M-BOM is configured to represent the assembly procedures of Pattern 1 and Pattern 2, while the BOP is kept coarse to reflect the context of early production planning. The defined processes in BOP included material allocation, tack welding, full welding, block turnover, and subsequent allocation, tack welding, and welding after turnover. Provided that a NAPA model is available, the preparation of BOM/BOP data and the execution of production simulations can be accomplished within a practical range of time and effort for real application.

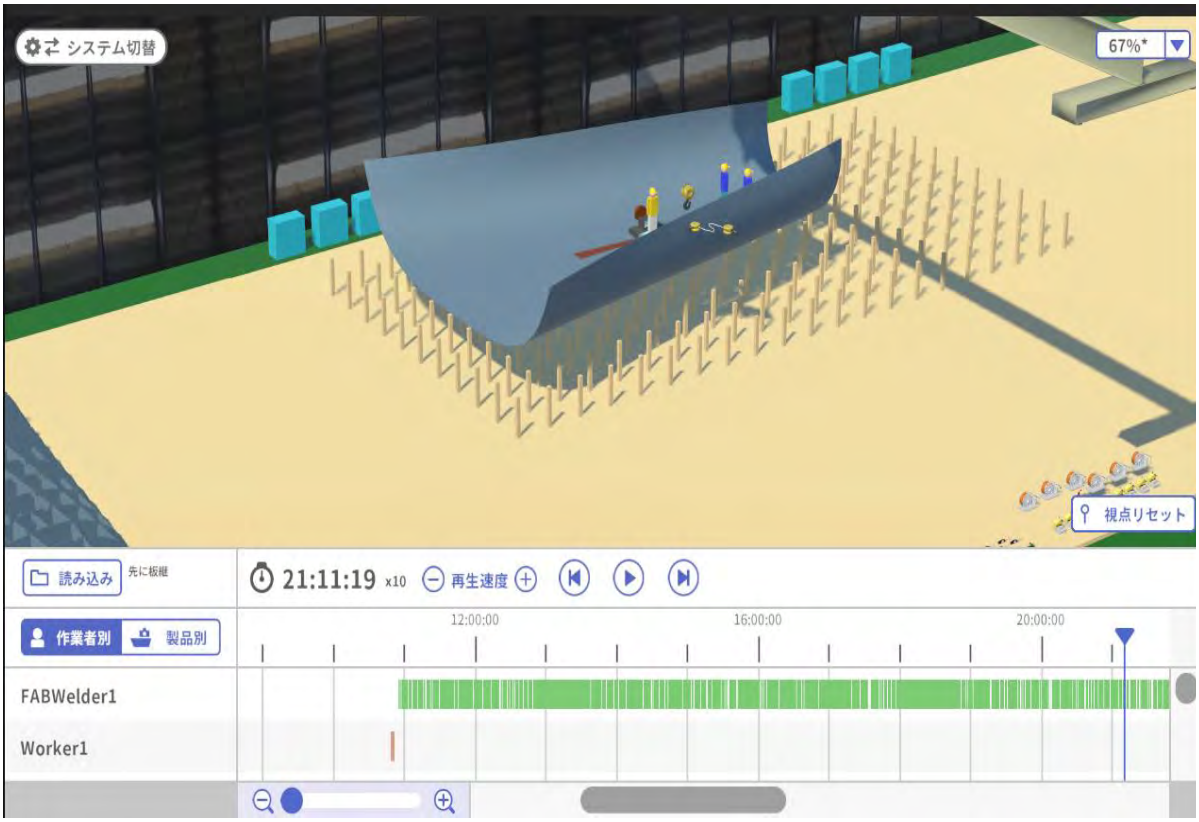
Production simulations were performed for both Pattern 1 and Pattern 2, and the differences between the two assembly methods could be reproduced within the virtual factory environment. The computational results provided detailed time histories of products and resources, including weld sequences, directions, and postures at the level of individual seams. The simulation provided quantitative estimates of task durations in minutes. When the same number of workers was assigned to each case, the overall assembly time did not differ significantly between the two methods. Specifically, once the inner-structure block was completed, the estimated duration was approximately 52 hours for Pattern 1 and 56 hours for Pattern 2. These values were obtained using a coarse BOP, reflecting the context of early production planning. Although the BOP used here is not sufficiently detailed to reproduce actual shop-floor operations faithfully, the results nonetheless demonstrate that, under consistent conditions, the simulation can provide meaningful approximations of man-hours, lead times, and relative differences between alternative assembly methods.

In addition to numerical results, the simulation highlighted practical aspects of each method. Pattern 1 requires safety measures during shell-plate attachment on the inner structure, particularly when workers are standing on the block during the operation, while Pattern 2 involves scaffolding, more complex fitting, and greater reliance on crane operations. In both cases, welding after turnover was confirmed to be feasible, with manholes providing sufficient access to confined areas, as the agent-based simulation explicitly demonstrated that tasks cannot be initiated if worker accessibility is not ensured.

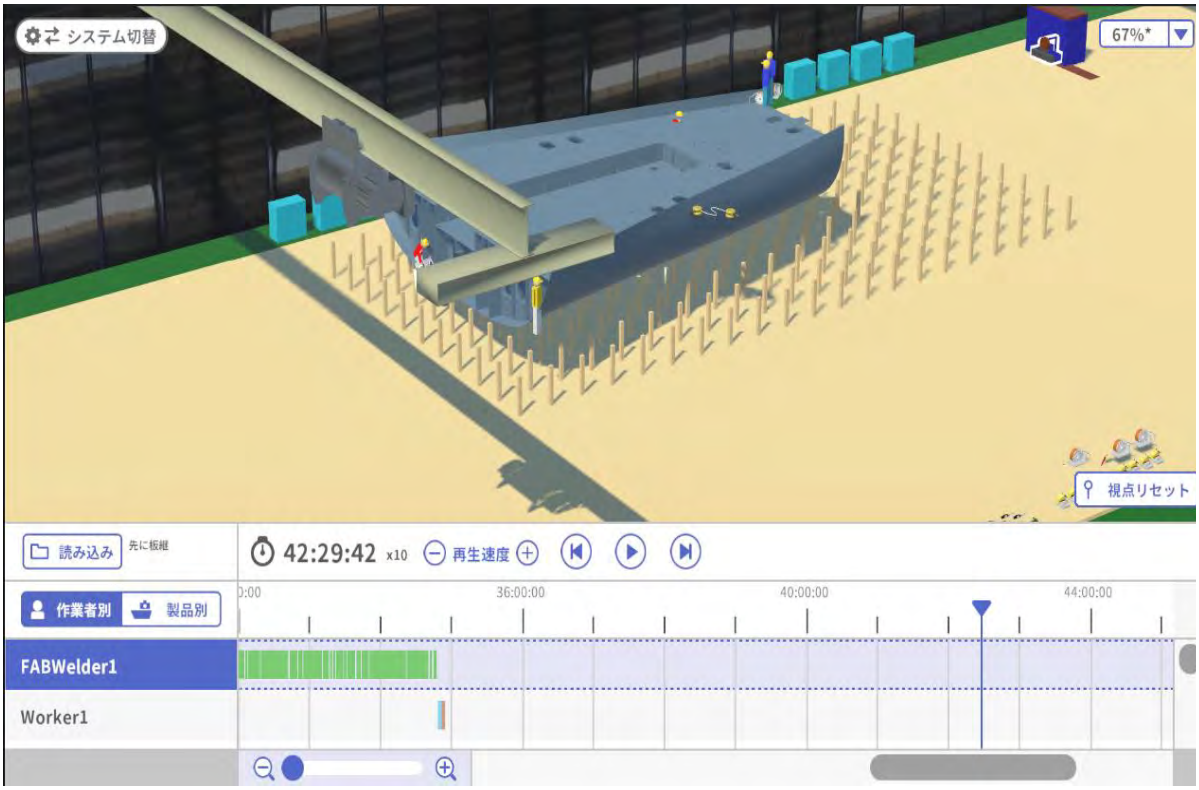
These findings demonstrate that the model-based approach, supported by production simulation, can provide both numerical evaluations and practical feedback to design and production at the early production planning stage. Unlike conventional practice—where planners relied on 2D drawings or 3D visualization tools and relied heavily on the imagined processes in their minds—this approach makes it easier to identify critical issues that would otherwise remain unnoticed until later stages. As decisions on block division and assembly methods are difficult to revise once established, this capability enables such critical decision-making more rationally, offering the potential to maximize efficiency from the first ship of a series.



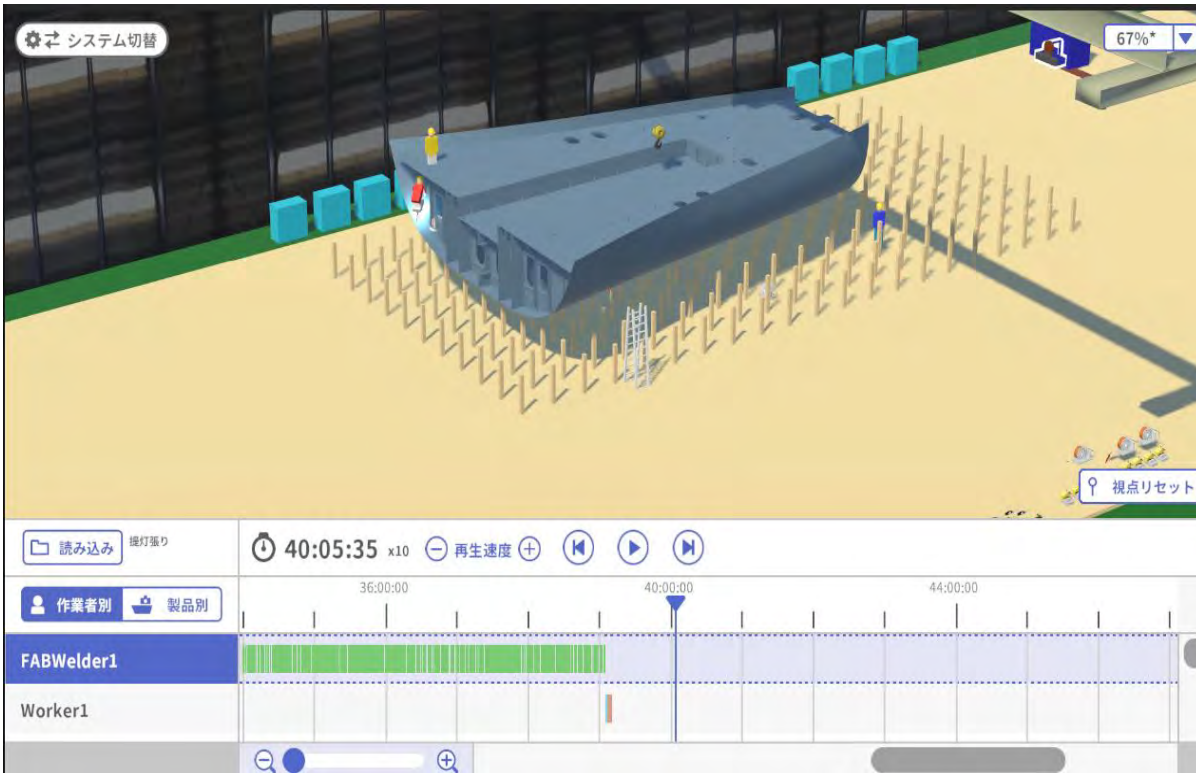
Pattern 1: Shell plate arrangement



Pattern 2: Shell plates preassembling



Pattern 2: Inner-structure block mounting.



Pattern1: Inner-structure block and shell plate welding.

Fig.7: Image of production simulation for block assembling.

5. Conclusions and Future Work

This study organized and examined the concept and practical implementation of the model-based approach in ship production planning. The main findings and contributions can be summarized as follows:

- The study clarified how production planning and evaluation can shift from experience-based practice toward model- and simulation-based approaches, emphasizing the role of early 3D models and production simulations in supporting rational decisions.
- Representative use cases at the early production planning stage were identified, showing how this approach can address current limitations and contribute to achieving higher efficiency from the first ship of a series.The study explored system implementation aspects, focusing on data flow between CAD and production simulation, and clarified the LOD of product and process data required to perform early-stage simulations effectively. While the demonstration was conducted using NAPA Steel, the approach is applicable to other CAD platforms.
- As a concrete demonstration, a case study on the assembly planning of an engine-room double-bottom block was conducted. This provided a tangible image of how simulation can be applied at the early production planning stage, including the required procedures, time frame, and system operation.

Looking ahead, several areas of further development are identified:

- Preparation of 3D models at an appropriate level of detail, enabling their use in early-stage production simulations.
- Preparation and automation of process data, including the development of methods for semi-automatic generation of M-BOM and BOP information.
- Validation of man-hour estimates for early assembly simulations (i.e., coarse BOP-based simulations) by comparison with actual man-hours or detailed assembly simulations, and development of methods to compensate for omitted ancillary tasks such as scaffolding and jig installations.
- Exploration of extended applications in which yard-specific constraints (e.g., availability of indoor turnover operations vs. outdoor facilities) and interdependencies among multiple blocks (e.g., allocation of Pattern 2 sub-assemblies to highly efficient flow production lines) are taken into account, thereby enabling more comprehensive optimisation of yard-specific assembly planning.
- Institutionalization of production planning practices that incorporate production simulation as a standard tool in shipyards.
- Creation of case studies demonstrating actual benefits of simulation-based production planning, to verify its practical effectiveness in reducing costs, improving scheduling accuracy, and enhancing safety.

In conclusion, while detailed and direct simulation has traditionally been applied to functional design analysis, this study emphasizes that similar approaches should now be extended to production planning and manufacturability evaluation. In this sense, a model-based approach for production simulation can be regarded as the “FEM of production”, leveraging existing 3D models prepared for FEM or class approval to enable early production planning. Advancing such model-based practices has the potential to drive innovation in shipbuilding by integrating design and production processes from the earlier stages of product development.

Acknowledgements

The authors would like to express their sincere gratitude to ASAKAWA Shipbuilding Co., Ltd. for providing the three-dimensional models and construction scenarios used in the demonstration presented in this study.

References

BURGGRÄF, P.; ADLON, T.; BEYER, M.; SCHÄFER, N.; FULTERER, J. (2022), *Enabling Individual Part Production Planning in Shipbuilding: Machine-Learning-Assisted Prediction of Production Times*, 21st COMPIT Conf., Pontignano, pp.110-119

MASUI, T.; SEPPÄLÄ, T.; SON, M.J.; HUOTARI, J. (2023), *Collaborative Single Model Design Platform for Ships of Tomorrow*, 22nd COMPIT Conf., Drübeck, pp.318-333

MATSUO, K.; MORISHITA, M.; HIYOKU, K.; SUEOKA, M.; TANIGUCHI, T.; TAKEZAWA, M. (2022), *A fundamental study on PLM system for shipbuilding*, Conf. Proc. Japan Soc. Naval Architects Ocean Eng. 34, pp.427-430

MATSUO, K.; MORISHITA, M.; TANIGUCHI, T. (2024), *Research on BOM data model for PLM system in shipbuilding*, 21st ICCAS Conf., Genova

MATSUO, K.; TANIGAWA, F. (2020), *Modelling of flexibility of shipbuilding work by production simulation*, 19th COMPIT Conf., Pontignano, pp.63-70

SAKAGAMI, M.; OTAGURO, T.; MASUI, T.; LIN, P.; SHIMAKAWA, Y.; HISANO, S. (2022), *Making 3D model-based approval a reality*, 20th ICCAS Conf., Yokohama

SON, M.J.; SEPPÄLÄ, T.; MERIKANTO, J.; AAE, O; ASTRUP, O.C. (2022), *Utilization of OCX as Part of 3D Model Based Approval in Ship Design Process*, 21st COMPIT Conf., Pontignano, pp.258-272

TAKEZAWA, M.; TANIGUCHI, T.; MATSUO, K.; HIRAKATA, M. (2020), *Fundamental study on highly precise production simulation for shipbuilding*, Conf. Proc. Japan Soc. Naval Architects Ocean Eng. 31, pp.177-183

TANIGUCHI, T.; MATSUO, K. (2024), *A study of process simulation based on a multi-agent system for shipbuilding*, 21st ICCAS Conf., Genova

TANIGUCHI, T.; TAKEZAWA, M.; MATSUO, K. (2022), Development of high precision process simulation based on multi-agent system for shipbuilding, J. Japan Soc. Naval Architects Ocean Eng. 36, pp.89-100

TANIGUCHI, T.; TANIGUCHI, T.; KIJI, N.; SADANO, M.; MATSUO, K. (2023), *Basic study on generation of Bill of Process and simulation for block assembly process*, Conf. Proc. Japan Soc. Naval Architects Ocean Eng. 36, pp.213-216

ZERBST, C. (2021), *Deming Cycle Enabled: A Digital Twin for Ship Production*, 20th COMPIT Conf., Mülheim, pp.122-129

ZERBST, C.; LUTZ, U.; DANETZKY, D. (2022), *Concept to Reality: Implementing a Digital Twin for Ship Production*, 21st COMPIT Conf., Pontignano, pp.336-342

Index by Authors

24th Conference on

Computer Applications and Information Technology in the Maritime Industries

COMPIT'27

Mülheim / Germany, 12-14 October 2027

Topics: Artificial Intelligence / CAX Techniques / Digital Twin / Simulations / Virtual & Augmented Reality / Robotics / Autonomous Technology

In Design, Production and Operation of Maritime Systems

Organiser: Volker Bertram (mailto:volker@vb-conferences.com)

Advisory Committee:

Marco Bibulii	CNR, Italy	Inno Gatin	Cloud Towing Tank, Croatia	Jialun Liu	WUT, China
Marcus Bole	CADMATIC, UK	Ken Goh	Knud E. Hansen, Australia	Kohei Matsuo	NMRI, Japan
Jean-David Caprace	COPPE, Brazil	Stefan Harries	Friendship Systems, Germany	Rodrigo Perez	Siemens, Spain
Thomas DeNucci	USCG, USA	Woo-Sung Kil	Korean Register, S. Korea	Ludmila Seppälä	NAPA, Finland
Henrique Gaspar	NTNU, Norway	Herbert Koelman	NHL, Netherlands	Carsten Zerbst	PROSTEP, Germany

Venue: The conference will be held in the "Die Wolfsburg" in Mülheim/Germany

Format: Papers to the above topics are invited and will be selected by a committee.

Deadlines:	anytime	Optional "early warning" of intent to submit paper
	10.6.2027	First round of abstract selection (approx. 1/2 slots)
	10.7.2027	Final round of abstract selection (remaining slots)
	15.9.2027	Payment due for authors
	20.9.2027	Final papers due (60 € surcharge for late submission)
Fees:	700 € / 400 €	regular / PhD student – early registration (by 1.8.2027)
	800 € / 450 €	regular / PhD student – late registration

Fees are subject to VAT
Fees include proceedings, lunches, coffee breaks and conference dinner

Sponsors: Tutech Innovation, (further sponsors to be announced)

Information: volker@vb-conferences.com