A Decision Support System for Autonomous Ship Trajectory Planning[⋆]

Melih Akdağ[a,*], Tom Arne Pedersen[b], Thor I. Fossen[a] and Tor Arne Johansen[a,**]

[a] *Norwegian University of Science and Technology, Department of Engineering Cybernetics, Trondheim, 7491, Norway*
[b] *DNV, Veritasveien 1, Høvik, 1363, Norway*

ARTICLE INFO

Keywords:
autonomous ship
consequence analysis
multi-objective optimization
multi-criteria decision-making
trajectory planning
decision support system

ABSTRACT

This study presents an approach to enhance human-machine collaboration in autonomous ship trajectory planning. A decision support system is developed, considering crucial environmental factors such as ocean currents, wind, and tidal information, alongside the integration of narrow channel geometry, squat effect, and Under-keel-clearance (UKC). The Dynamic Consequence Analysis (DCA) risk assessment method is utilized to establish dynamic safety domains, incorporating ship maneuvering characteristics and potential failure scenarios. The Multi-objective Particle Swarm Optimization (MOPSO) algorithm is then employed to generate alternative trajectories, optimizing five objective functions: minimizing safety domain violation, consecutive speed changes, path length, deviation from the initial plan, and deviation from the initial estimated time of arrival. Furthermore, two Multi-criteria Decision Making (MCDM) methods, Multi-Objective Optimization by Ratio Analysis (MOORA) with user preferences, and the Entropy Weight Method (EWM) with automatic weight allocation, are employed to rank alternative solutions from the Pareto front. Finally, a clustering method is employed on the Pareto front solutions. The outcomes that serve as representatives for these clusters are then merged with the highest-rated alternative solutions from MCDM methods. This combined set forms the basis for the final decision-making process carried out by the operator. While dynamic obstacles are not considered in this study, evaluation across three scenarios demonstrates the effectiveness of the proposed decision support system for trajectory planning.

1. Introduction

While there have been notable advancements in autonomous ship technologies, there are concerns regarding their alignment with existing regulatory frameworks (IMO, 2021). The slow-paced nature of regulation-making processes, typically characterized by careful considerations of safety and operational implications, contrasts with the rapid evolution of autonomous systems (Negenborn, Goerlandt, Johansen, Slaets, Banda, Vanelslander and Ventikos, 2023). The International Maritime Organization (IMO) has introduced four autonomy levels to define the progressive integration of autonomous functionalities in ships (IMO, 2021). However, before reaching autonomy levels 3 or 4, which involve a high degree of vessel autonomy, the feasibility and practicality of implementing periodically unmanned bridges and decision support systems for seafarers should be explored. These intermediate stages offer valuable opportunities to learn from the experiences gained, refine existing and emerging technologies, and enhance safety measures in the transition toward fully autonomous shipping.

Autonomous ship navigation, when limited to trajectory planning, accounts for determining a path for the ship to follow from its current position to its desired destination while considering the ship's dynamics, environmental conditions, safety requirements, economy, operational constraints, and static and dynamic obstacles. The static obstacles encompass elements such as land, shores, and depths, while dynamic obstacles comprise other vessels and floating objects encountered during the ship's route. The trajectory planning process aims to achieve the goal of reaching a destination while ensuring Collision and Grounding Avoidance (CAGA) with obstacles as well as optimizing efficiency objectives along the way. Optimization techniques are essential to tackle the high complexity of the planning problem and find the most optimal paths and maneuvers in real time. Various algorithms, such as

[⋆] This work was supported by the Center for Research-based Innovation AutoShip, project number 309230.

[*] Corresponding author

[**] Principal corresponding author

melih.akdag@ntnu.no (M. Akdağ); tom.arne.pedersen@dnv.com (T.A. Pedersen); thor.fossen@ntnu.no (T.I. Fossen); tor.arne.johansen@ntnu.no (T.A. Johansen)

ORCID(s): 0000-0002-6879-1799 (M. Akdağ)

A*, Rapidly-exploring Random Trees, Artificial Potential Field, Genetic Algorithms, Model Predictive Control, and more, are extensively employed for autonomous ship trajectory planning (Öztürk, Akdağ and Ayabakan, 2022). An optimization problem contains the core elements of the objective function, decision variables, and, if applicable, constraints. The objective function enables us to measure a solution's performance among other solutions. Decision variables set the range of choices available which have an impact on the solution performance and constraints set limits to be satisfied. The majority of trajectory planning applications consider single-objective optimization due to its simplicity and straightforward nature. In single-objective optimization, the objective function represents a single goal to be maximized or minimized, making it easier to define and implement. However, in many scenarios, there are often multiple criteria or objectives that need to be simultaneously optimized, but these objectives are typically competing with each other. Multi-objective Optimization (MOO) is a powerful mathematical approach that deals with finding the best possible solutions for problems that involve multiple conflicting objectives. MOO aims to identify a set of solutions that represents a trade-off between the conflicting objectives, known as the Pareto front (Ngatchou, Zarei and El-Sharkawi, 2005). The Pareto front consists of solutions that are not dominated by any other solution in terms of all objectives simultaneously. Dominance refers to one solution being superior to another in at least one objective and not worse in any other objective. Therefore, a solution that lies on the Pareto front represents an optimal compromise, where improving one objective would require sacrificing performance in another objective. Different solutions on the Pareto front provide different trade-offs between the objectives, enabling decision-makers to make informed decisions based on their priorities and constraints. Therefore, the MOO presents a promising approach for creating a decision support system for trajectory planning that can offer operators multiple solutions.

Acronyms

ACO	Ant Colony Optimization.
AHP	Analytic Hierarchy Process.
APF	Artificial Potential Field.
CAGA	Collision and Grounding Avoidance.
COLREG	Convention on the International Regulations for Preventing Collisions at Sea.
CRITIC	CRiteria Importance Through Inter-criteria Correlations.
DCA	Dynamic Consequence Analysis.
DNN	Deep Neural Network.
DP	Dynamic Positioning.
EA	Evolutionary Algorithms.
ECDIS	Electronic Chart Display and Information System.
ENC	Electronic Navigation Chart.
ETA	Estimated Time of Arrival.
EWM	Entropy Weight Method.
GA	Genetic Algorithm.
IMO	The International Maritime Organization.
MCDM	Multi-criteria Decision Making.
MOO	Multi-objective Optimization.
MOORA	Multi-Objective Optimization by Ratio Analysis.
MOPSO	Multi-objective Particle Swarm Optimization.
MPC	Model Predictive Control.
PRM	Probabilistic Roadmap Method.
PSO	Particle Swarm Optimization.
RL	Reinforcement Learning.
RRT	Rapidly Exploring Random Tree.
TOPSIS	Technique for Order of Preference by Similarity to Ideal Solution.
UKC	Under-keel-clearance.

1.1. Related works

Path planning can be broadly classified into global and local algorithms based on their spatial coverage (Öztürk et al., 2022). Global path planning algorithms typically account for static obstacles and environmental forces to determine a viable path between two positions, incorporating defined objectives such as preventing grounding, optimizing path length, and enhancing energy efficiency. In contrast, local path planning algorithms focus on dynamic obstacles and cover smaller areas compared to their global counterparts. Due to the presence of dynamic obstacles, local path planning algorithms often require short execution times to operate in real-time scenarios, as they may need to run multiple times during a scenario.

However, this classification is not universally applicable, as there are studies that integrate both local and global path planning algorithms. This integration can be achieved through the development or implementation of algorithms with short execution times or by designing a hierarchical architecture. Such a hierarchical approach might involve utilizing a slower global planner to generate an initial feasible trajectory and a faster local planner to dynamically update the trajectory in response to changing obstacles and environmental conditions.

The landscape of path planning algorithms is diverse, encompassing various approaches such as environmental modeling algorithms (e.g., Voronoi graph and Visibility graph), deterministic methods (e.g., Artificial Potential Field (APF), graph search algorithms like Dijkstra and A*, and local reactive planners like Dynamic Window, Velocity Obstacle, and Model Predictive Control (MPC)), non-deterministic methods (e.g., sampling-based search methods like Probabilistic Roadmap Method (PRM), Rapidly Exploring Random Tree (RRT)/RRT*, and population-based stochastic algorithms like Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Evolutionary Algorithms (EA), Genetic Algorithm (GA)), and other artificial intelligence methods (e.g., Deep Neural Network (DNN), Reinforcement Learning (RL)). For an in-depth exploration of these methods, readers are referred to recent literature reviews such as Öztürk et al. (2022); Burmeister and Constapel (2021); Huang, Chen, Chen, Negenborn and Van Gelder (2020); Vagale, Oucheikh, Bye, Osen and Fossen (2021). In the rest of this section, we provide a list of related works that have had an influence on the development of this study by their implementation of MOO, Multi-criteria Decision Making (MCDM), risk analysis methods, and database usage.

A widely recognized study utilizing MOO on autonomous ship trajectory planning has been developed by Szlapczynski and Szlapczynska (2012). The study proposes a method for multi-ship trajectory planning using evolutionary computing and MOO, which can be used to find safe and efficient trajectories for multiple ships in a congested environment using non-deterministic methods. The study covers several MOO EA and considers both the dynamic and static obstacles but does not demonstrate a method to choose a solution from the Pareto front.

Another non-deterministic method is proposed by Lazarowska (2017a) with a multi-criteria ACO algorithm for collision avoidance of ships with both static and dynamic obstacles. The study considers collision risk, path length, path smoothness, and the Convention on the International Regulations for Preventing Collisions at Sea (COLREG) but lacks environmental forces. The proposed algorithm finds possible solutions and the weighted objectives MCDM method is applied to choose a final solution. Same author also proposes a deterministic path planning algorithm as a decision support system in Lazarowska (2017b). The method utilizes comparing alternative trajectories from a database while considering static, dynamic obstacles, COLREG, and maneuvering times for course changes. Deterministic nature and the low execution time are the advantages of the approach however the method can be improved with implementing environmental forces and their effect on the decisions.

Hu, Naeem, Rajabally, Watson, Mills, Bhuiyan, Raeburn, Salter and Pekcan (2019) proposes a MOO for path planning of autonomous ships that partially adheres to the COLREG. The proposed approach is implemented using the Multi-objective Particle Swarm Optimization (MOPSO) algorithm. The study incorporates a hierarchical sorting rule which favors course/speed change over other objectives. However, the study does not consider environmental forces, aims at local trajectory planning for short ranges, and does not propose a method to evaluate the Pareto front solutions.

Blindheim and Johansen (2022) proposes a method for dynamic risk-aware path following for autonomous ships using PSO. The method discretizes the original path into a sequence of waypoints and considers environmental forces. The method is evaluated in a simulation environment and is able to find a new path that minimized the risk of grounding while still satisfying the constraints of the original path. However, a single objective function consisting of a weighted summation of different objectives is used in the study which outputs a single solution.

Except in Blindheim and Johansen (2022), the studies mentioned before do not cover environmental forces and their effects on the ship's trajectory. Additionally, technical failures of ship systems and their consequences on the ship's trajectories are not taken into account. Thieme, Rokseth and Utne (2023) defines several approaches for integrating risk analysis methods with autonomous vehicles' control systems such as using the output of risk models as input to decision-making and optimization algorithms. Furthermore, Fossdal (2018) fills the gap by implementing an Dynamic Consequence Analysis (DCA) for risk assessment of autonomous ship navigation. The study considers environmental forces and a set of technical failure modes to mitigate collision and grounding risks. However, the study is limited to defining the consequence analysis without the implementation of a path planning algorithm or incorporation of the maneuvering characteristics of ships such as turn circles or stopping distance. Similarly, Rokseth, Haugen and Utne (2019) uses the DCA with Systems-Theoretic Process Analysis risk assessment method to verify the safety of autonomous ships. Blindheim, Rokseth and Johansen (2023) extends the study Blindheim and Johansen (2022) by including machinery system operation modes into the PSO to include several machinery configurations and their contribution to trajectory planning with time-to-grounding risk calculations.

1.2. Contributions

To further address these issues, this study utilizes the features from the previously mentioned approaches in a decision support system for trajectory planning of ships considering the static obstacles, DCA, MOO, and MCDM concepts. Based on this objective, the following research questions will be addressed in the study:

Question 1: In what ways can autonomous ship navigation be enhanced through the integration of information about environmental forces, navigable waters, maneuvering characteristics, and potential failure modes?

Question 2: What advantages and obstacles are associated with the utilization of a decision support system based on MOO for autonomous ship navigation?

Question 3: What strategies can be employed to foster collaboration between autonomous systems and operators during the decision-making process of ship navigation?

The contributions of the study are listed as

i. Exploiting MOO in a decision support system to propose alternative solutions to operators.

ii. Utilizing MCDM and clustering methods to prune and rank alternative solutions in the Pareto front.

iii. Using the DCA concept for creating ship safety domain geometry to be used in trajectory planning.

iv. Incorporating the UKC concept in autonomous ship navigation.

v. Introducing a method to evaluate narrow channel geometry to use in the UKC concept in autonomous ship navigation.

The rest of the article is organized as follows: Section 2 presents the methodology, beginning with mathematical models of ship and the considered environmental forces. Then, the topic of navigational charts and chart depths is introduced. Following that, the dynamic draught of a ship and a method for evaluating narrow channel geometry is explained, along with an evaluation of the concept of unde-keel-clearance. The maneuvering characteristics of a ship and the DCA method are then introduced, before the usage of MOO, MCDM, and clustering methods are explained. Section 3 presents the simulation results from three different scenarios that demonstrate the capabilities and features of the method. Section 4 of the paper revisits the research questions and discusses both the merits and drawbacks of the proposed methodology. Additionally, the section highlights potential directions for future research. Lastly, Section 5 serves as the conclusion of the paper and offers a summary of the findings.

2. Methodology

Fig. 1 presents the overall architecture of the proposed method with phases, modules, and their relations. The final goal of the proposed method is to develop a decision-support system for optimizing an initial trajectory plan considering environmental forces, ship characteristics, possible failures, and multiple objectives, e.g. path length, grounding risk, and Estimated Time of Arrival (ETA). The proposed method divides the problem into four phases such as initialization, preparation, optimization, and decision-making. The initialization phase contains the information that is already available such as Electronic Navigation Chart (ENC) data, static and dynamic ship states, initial trajectory plan, mathematical models of the ship, ocean current and wind, and up-to-date water level information. The preparation phase uses the existing information and updates the map, generates templates for risk assessment, and asks users their preferences related to objectives that are considered in the decision-making phase. The optimization phase calculates alternative trajectory plans with MOO methods. During the decision-making phase, the user is presented with various alternative solutions that are evaluated and ranked, ultimately allowing them to make a final decision. Following subsections explain the methods used in the modules that make up the overall architecture of the study.

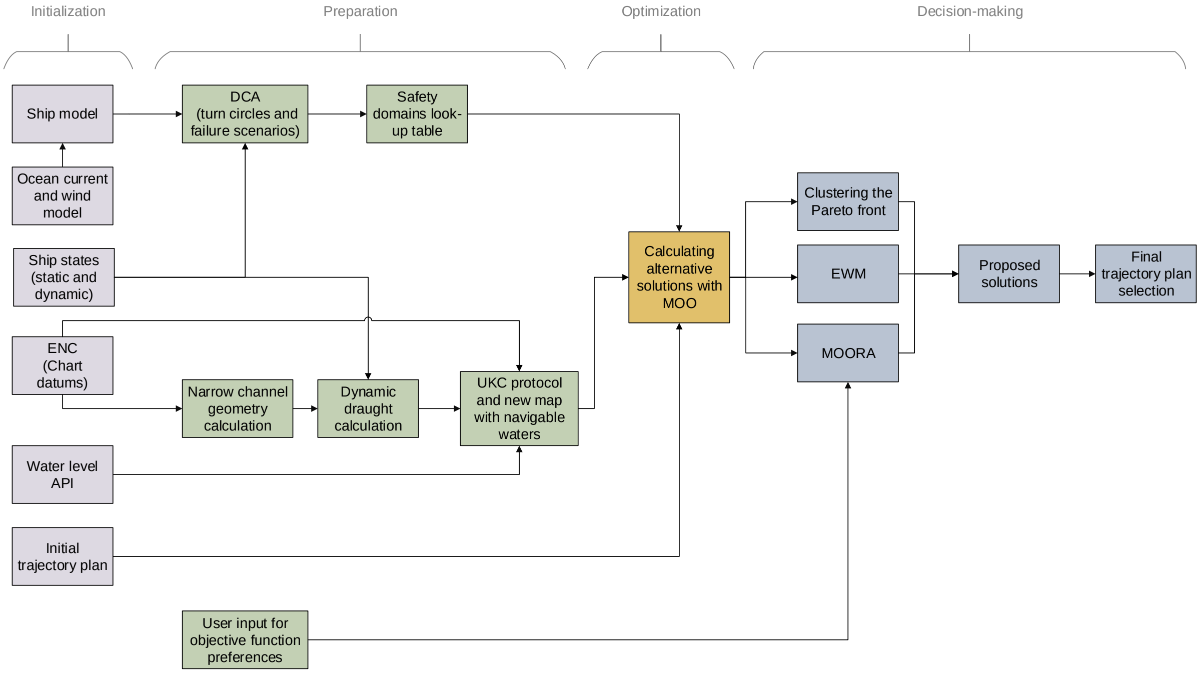

Initialization
Preparation
Optimization
Decision-making
Ship model
DCA (turn circles and failure scenarios)
Safety domains look-up table
Ocean current and wind model
Ship states (static and dynamic)
ENC (Chart datums)
Narrow channel geometry calculation
Dynamic draught calculation
UKC protocol and new map with navigable waters
Water level API
Initial trajectory plan
User input for objective function preferences
Calculating alternative solutions with MOO
Clustering the Pareto front
EWM
MOORA
Proposed solutions
Final trajectory plan selection

Figure 1: The overall architecture of the system proposed in the study.

2.1. Mathematical Ship Model

A 3-DOF ship maneuvering model from Fossen (2021) is implemented both for use in the model-based control system and for simulation:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)\boldsymbol{\nu} \tag{1a}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}}_{\boldsymbol{r}} + \boldsymbol{C}(\boldsymbol{\nu}_{\boldsymbol{r}})\boldsymbol{\nu}_{\boldsymbol{r}} + \boldsymbol{D}(\boldsymbol{\nu}_{\boldsymbol{r}})\boldsymbol{\nu}_{\boldsymbol{r}} = \boldsymbol{\tau} + \boldsymbol{\tau}_{\boldsymbol{wind}} \tag{1b}$$

$$\boldsymbol{R}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1c}$$

where $\boldsymbol{\eta} = [x, y, \psi]^{\mathrm{T}}$ contains the position and heading angle in the Earth-fixed Cartesian coordinate frame, $\boldsymbol{\nu} = [u, v, r]^{\mathrm{T}}$ is the surge and sway velocities, and yaw rate, decomposed in the body-fixed coordinate frame, $\boldsymbol{\nu_c} = [u_c, v_c, 0]^{\mathrm{T}}$ is the ocean current velocity vector, and $\boldsymbol{\nu_r} = \boldsymbol{\nu} - \boldsymbol{\nu_c}$ is the relative velocity vector. $\boldsymbol{\tau}$ is the vector of generalized forces generated by the control inputs and $\boldsymbol{\tau_{wind}}$ is the vector of generalized wind forces. The matrices $\boldsymbol{M}$, $\boldsymbol{C}(\cdot)$, $\boldsymbol{D}(\cdot)$ represent inertia, Coriolis-centripetal, and damping, respectively, and the rotation matrix $\boldsymbol{R}(\cdot)$ is used to transform the body-fixed frame to the earth-fixed frame. Heading control is achieved by a PID controller as

$$\tau_N = \tau_{FF} - K_p\tilde{\psi} - K_d\tilde{r} - K_i\int_0^t \tilde{\psi}(\tau)d\tau \tag{2a}$$

$$\tau_{FF} = m(\dot{r}_d + \frac{1}{T}r_d) \tag{2b}$$

where $\tilde{\psi} = \psi - \psi_d$, $\tilde{r} = r - r_d$, K_p, K_d, and K_i are proportional, derivative, and integral gains, and τ_{FF} is the reference feed-forward term. Surge speed control is obtained with a PI controller as

$$\tau_X = -K_p\tilde{u} - K_i\int_0^t \tilde{u}(\tau)d\tau \tag{3}$$

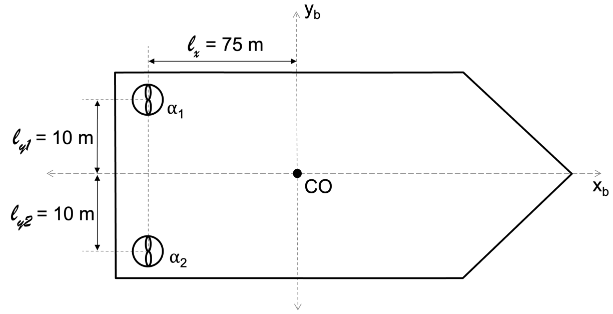


Figure 2: The diagram of thrusters. α_i is the azimuth angle of thruster i and l_{x_i} and l_{y_i} are the thruster's perpendicular distances from the coordinate origin (CO).

where $\tilde{u} = u - u_d$. Controller gains are derived by the pole placement method as

$$K_p = m\omega_n^2 \tag{4a}$$
$$K_d = 2\zeta\omega_n m \tag{4b}$$
$$K_i = \frac{\omega_n}{10} K_p \tag{4c}$$

where ω_n is the desired natural frequency and ζ is the desired relative damping ratio. Control forces are combined in a generalized control forces vector with $\boldsymbol{\tau} = [\tau_X, 0, \tau_N]^{\mathrm{T}}$.

The ship used in the study and its thrust configuration is presented in Fig 2. The ship has two azimuth thrusters used for main propulsion and steering. Control allocation is needed to distribute generalized control forces ($\boldsymbol{\tau}$) to the individual thruster as control inputs (u_i where i represents a single thruster). The control allocation is achieved by

$$\boldsymbol{u}_e = \boldsymbol{B}_e^{-1}\boldsymbol{\tau} \tag{5a}$$
$$\boldsymbol{B}_e = \boldsymbol{K}_e\boldsymbol{T}_e \tag{5b}$$
$$\boldsymbol{T}_e = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ -\ell_{y_1} & -\ell_{x_1} & \ell_{y_2} & -\ell_{x_2} \end{bmatrix} \tag{5c}$$

where $\boldsymbol{u}_e = [u_{1_x}, u_{1_y}, u_{2_x}, u_{2_y}]^{\mathrm{T}}$ is the extended thrust vector, $\boldsymbol{K}_e = \mathrm{diag}\{K_1, K_1, K_2, K_2\}$ is the extended diagonal force coefficient matrix, $\boldsymbol{T}_e$ is the extended thrust configuration matrix. Azimuth control inputs can be calculated from the extended thrust vector with

$$u_i = \sqrt{u_{i_x}^2 + u_{i_y}^2} \tag{6a}$$
$$\alpha_i = \mathrm{atan2}(u_{i_y}, u_{i_x}) \tag{6b}$$

The desired setpoints for heading ψ_d and yaw rate r_d are calculated by the Adaptive Line-of-Sight guidance law from Fossen (2023):

$$\psi_d = \pi_p - \hat{\beta}_c - \tan^{-1}(\frac{y_e^p}{\Delta}) \tag{7a}$$
$$\dot{\hat{\beta}}_c = \gamma \frac{\Delta}{\sqrt{\Delta^2 + (y_e^p)^2}} y_e^p \tag{7b}$$
$$\pi_p = \mathrm{atan2}(y_{wp+} - y_{wp}, x_{wp+} - x_{wp}) \tag{7c}$$
$$y_e^p = \sin(\pi_p)(x - x_{wp}) - \cos(\pi_p)(y - y_{wp}) \tag{7d}$$
$$r_d = \dot{\psi}_d \tag{7e}$$

where π_p is path-tangential angle, $\hat{\beta}_c$ is the parameter estimate, y_e^p is cross-track-distance, Δ is predefined look-ahead distance, γ is the adaptation gain, and $(x_{wp}, y_{wp}), (x_{wp+}, y_{wp+})$ are current and next waypoints. Lastly, the trajectory planning algorithm, i.e., the MOO algorithm will output the desired surge speed (u_d) for each waypoint leg.

2.2. Implementation of environmental forces

A trajectory planning algorithm should take into account the effects of environmental forces since the maneuvering characteristics of a ship are influenced by ocean currents, waves, and wind forces, especially for low ship speeds. Furthermore, environmental factors can be utilized to compute alternative paths that prioritize fuel efficiency or passenger comfort. This study includes ocean currents and wind since they have a dominant effect on the ship's trajectory. The wave forces are not included since the ocean current encompasses wave-induced drift, and the water level concept in subsection 2.3 contains sea state (which is connected with wave spectrum parameters) and tidal information and their impact on vertical space, leading to alterations in depth. The effects of the ocean current and wind are especially apparent in dynamic consequence analysis which will be demonstrated later.

To determine the wind forces on a ship, we calculate the relative wind speed V_{rw} and angle of attack γ_{rw}, which both vary depending on the ship's heading and speed.

$$V_{rw} = \sqrt{u_{rw}^2 + v_{rw}^2} \tag{8a}$$

$$\gamma_{rw} = -\text{atan2}(v_{rw}, u_{rw}) \tag{8b}$$

$$u_{rw} = u - u_w \tag{8c}$$

$$v_{rw} = v - v_w \tag{8d}$$

$$u_w = V_w \cos(\beta_{V_w} - \psi) \tag{8e}$$

$$v_w = V_w \sin(\beta_{V_w} - \psi) \tag{8f}$$

where V_w and β_{V_w} are wind speed and direction. The wind model of Blendermann (1994) is used and the generalized wind forces are calculated by

$$\boldsymbol{\tau_{wind}} = \frac{1}{2}\rho_a V_{rw}^2 \begin{bmatrix} C_X(\gamma_{rw}) A_{F_w} \\ C_Y(\gamma_{rw}) A_{L_w} \\ C_N(\gamma_{rw}) A_{L_w} L_{oa} \end{bmatrix} \tag{9}$$

where C_X, C_Y, and C_N are wind coefficients, A_{F_w} and A_{L_w} are the ship's frontal and lateral projected areas to the wind, and L_{oa} represents the length overall. Equations for calculating wind coefficients (C_X, C_Y, C_N) can be found in Fossen (2021).

2.3. Chart datum, tidal information, and water level

ENC are widely used in maritime navigation in parallel with printed charts. Integration with ship systems such as the global navigation satellite system, gyrocompass, and RADAR enables seafarers to visualize navigable waters, planned routes, static and dynamic obstacles, etc. on the Electronic Chart Display and Information System (ECDIS) in real-time. Bathymetric depths displayed on ECDIS are important for seafarers to plan their voyages beforehand while echo sounders allow seafarers to cross-check the actual depths beneath the ship. Depth information displayed on both the printed and electronic charts are not always the real depths of the region due to the effects of the tides, currents, and weather states. Navigational charts use a reference level called the chart datum to demonstrate the water depths. The chart datum can be derived from mean sea level or more commonly tidal phases such as lowest astronomical tide or mean lower low water. Seafarers use chart datums and tidal charts in combination to plan and ensure safe passages. Additionally, astronomical tides can cause local currents which affect ship speed and maneuverability. In addition to tides, weather and sea states such as strong winds, air pressure, and waves can cause changes in water levels.

It is essential to convey this knowledge and practice in autonomous ship navigation to have more efficient and safe trajectory planning. In this study, open-source ENC data from the Norwegian Mapping Authority contains depth data projected in EUREF89 UTM zone 33N is used. An open-source library, the Seacharts (Blindheim and Johansen, 2021) is used to create 2D polygons from the ENC data for trajectory planning and simulation purposes. The depth information in the ENC referenced to nautical chart zero level. The water level prediction API from the Norwegian Mapping Authority is used to include the effects of tides and weather on top of the chart datum as water level (WL) value in centimeters.

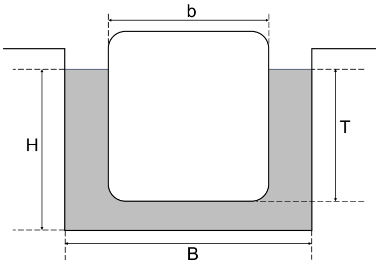


Figure 3: Ship in a canal. H is water depth, B is the width of the canal, T is the ship's static draught, and b is the ship' breadth.

2.4. Static and dynamic draughts of ships

Once the water depths of the region are known, the ship's draught before the voyage can be extracted from it to ensure safe passage of the trajectory. However, a ship's draught is a dynamic value and is affected by ship maneuvers, environmental forces, and the topography of the area it is sailing. When a ship is in motion, particularly at higher speeds, it experiences hydrodynamic forces that can cause the vessel to sink deeper into the water. This effect is primarily due to the interaction between the ship's hull and the surrounding water and is called the squat effect. For smaller and slower ships the squat effect can be within centimeters but for bigger and faster ships, the squat effect can be in a range of 1-2 meters and is an important phenomenon to consider in trajectory planning. The downward motion is greater, especially in narrow channels. Seafarers consider the squat effect in passage planning and there are several methods to calculate it. In this study, Eq. 10 from Barrass and Derrett (2011) is used to calculate the maximum squat effect in both narrow channels and open water.

$$\delta_{max} = \begin{cases} \frac{C_b V_k^2}{100}, & \text{if open waters and } 1.1 \leq \frac{H}{T} \leq 1.4 \\ \frac{C_b V_k^2}{50}, & \text{if narrow channels and } 0.1 \leq S \leq 0.25 \end{cases} \tag{10a}$$

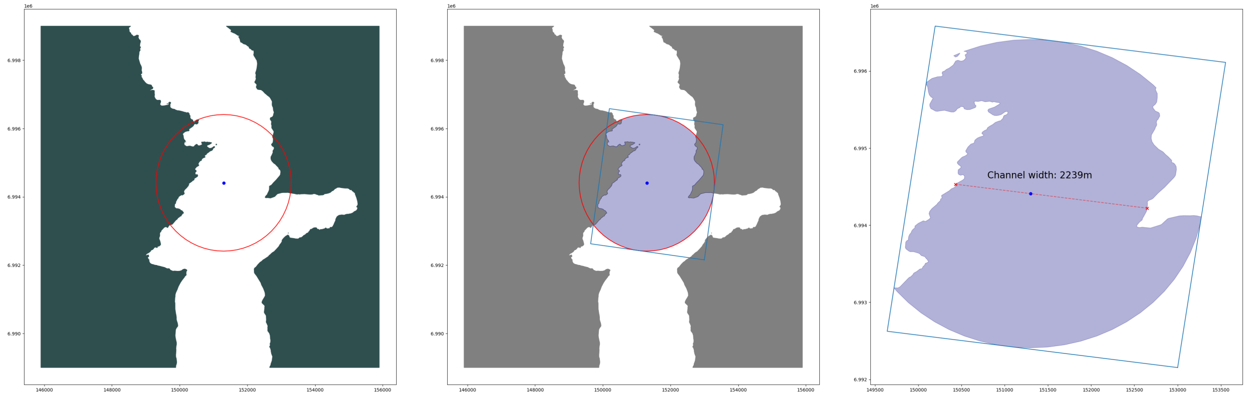
$$S = \frac{bT}{BH} \tag{10b}$$

where δ_{max} is maximum squat, C_b is block coefficient, V_k is ship's speed relative to the water or current, H is water depth, T is ship's static draught, b is ship's breadth, B is the width of the channel, and S is blockage factor and Fig. 3 illustrates the parameters.

2.5. Evaluating narrow channel geometry

The Eq. 10 requires evaluation of the geometry of the area where the ship is sailing. Especially, the squat effect is bigger in a canal or narrow channel where the width and depth of the passage are small compared to the ship's breadth and draught values. In the maritime domain, a narrow channel refers to a restricted waterway or passage that is typically narrower than the surrounding navigable waters. It is a designated route that enables ships to transit through areas with limited width, often due to geographical features such as land formations, bridges, or man-made structures. Although, the specific width that qualifies as a narrow channel can differ between waterways and is not explicitly defined. According to court cases, the width of a passage as much as 2 nautical miles (3704 meters) can be considered a narrow channel (Cockcroft and Lameijer, 2003) and is used as a threshold value to differentiate a narrow channel from open water in this study.

The width of a channel can be calculated with the help of the RADAR onboard ships. In this study, we propose a geometric method to approximately calculate the width of the channel using the position of the ship, the polygon data of the environment, and trigonometric functions. If the course of the ship is in alignment with the channel direction, the width of the channel can be found by calculating the perpendicular distances from the ship to each side of the channel. Since this is not always the case, a practical solution is applied. First, a circle with a defined radius centered in the ship's position is created. Then the difference between the circle and the land polygon is extracted if they intersect.



(a) Creating the circle around the ship position.

(b) Extracting the intersection polygon of circle and land polygons. Fitting the minimum rotated rectangle around the intersection polygon.

(c) Calculating the perpendicular distances from the ship position using the channel direction.

Figure 4: The steps of the narrow channel evaluation method.

Later, a minimum bounding rectangle that contains the intersection polygon is found. Minimum rotated bounding box algorithms differ from minimum bounding box algorithms since the rectangles found by minimum bounding box algorithms are in alignment with the axis but the rectangles found by the minimum rotated bounding box are in alignment with the polygon orientation. The direction of the channel is approximately in alignment with the rotation angle of this rectangle. The width of the channel can be found by calculating the perpendicular distances from the ship to each side of the channel using the channel direction angle. Fig. 4 illustrates the steps of the method with an example. It is important to define a proper radius value for the circle and 5000 meters is chosen as the radius in this study. Shapely, a computational geometry library developed for spatial analysis and geographic information system applications, is used for creating and modifying geometric shapes and performing spatial operations (Gillies, van der Wel, Van den Bossche, Taves, Arnott, Ward et al., 2023).

2.6. Navigable waters and under-keel-clearance

After updating the chart datum with water level and calculating the dynamic ship draught considering the squat effect, navigable waters and areas with grounding risk can be extracted from the map. Seafarers use the Under-keel-clearance (UKC) concept to evaluate if a passage is safe without grounding. UKC refers to the vertical distance between the lowest point of the ship's keel and the seabed. Ship masters and port authorities define the UKC management strategies considering the ship's maneuvering characteristics and dynamic draught, chart datum, tide, and sea states. There are different UKC management strategies such as defining a fixed minimum UKC (15% of the draught), UKC tables for different ship classes and tide information, or UKC software. Eq. 11 presents the minimum UKC calculation.

$$T_{dyn} = T + \delta_{max} \tag{11a}$$

$$UKC_{min} = C_{ukc} T_{dyn} - WL \tag{11b}$$

where T and T_{dyn} are ship's static and dynamic draughts, δ_{max} is maximum squat effect, WL is the water level value in meters, and C_{ukc} is UKC management protocol coefficient such as $C_{ukc} = 1.5$ in this study. Once the minimum UKC depth is defined, depth polygons smaller than the minimum UKC value are filtered and named grounding polygons. This updated map goes into the optimization phase in Fig. 1 to be used in the trajectory planning algorithm.

2.7. The maneuvering characteristics

Maneuvering characteristics refer to the inherent attributes and capabilities of a ship that determine its ability to change direction, turn, stop, and navigate effectively in various conditions. Ship maneuverability is a crucial aspect of maritime operations, and it is commonly assessed through the analysis of turn circles, Kempf's zigzag maneuver, pull-out maneuver, Dieudonné's spiral maneuver, Bech's reverse spiral maneuver, and stopping trials (Fossen, 2021).

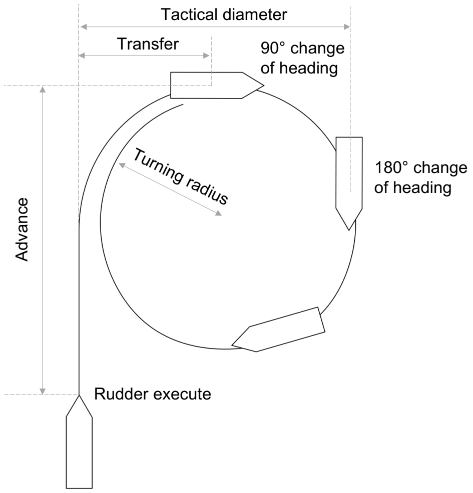


Figure 5: Turning circle diagram of a ship.

The turning circle trial is one of the standard procedures for evaluating a ship's maneuverability by measuring steady turning radius, tactical diameter, advance, and transfer values by applying constant speed and rudder angle. Turning circle definitions are presented in Fig. 5. Stopping trials contain crash stop and low-speed stopping trials. Crash stop trials evaluate a ship's emergency stopping capabilities at high speeds including reverse thrust, while low-speed stopping trials assess its stopping performance during routine navigation and docking maneuvers at lower velocities. Additionally, the COLREG rule 6 incorporates considerations of turning circle and stopping characteristics in safe speed evaluation (IMO, 1972). In this study, only the turning circle characteristics are incorporated for different ship speeds in dynamic consequence analysis.

2.8. Dynamic Consequence Analysis

Dynamic Consequence Analysis (DCA) refers to the process of assessing and evaluating the potential outcomes and effects of failures or accidents in real-time or near real-time. By doing so, it aims to provide timely information and insights that can support decision-making, emergency response, and mitigation efforts. The first step in DCA is to define failure modes, which involves identifying the various potential operational failures or incidents that may occur in a given system or process. These failure modes can range from equipment malfunctions to external events. Once the failure modes are defined, the next step is determining the simulation times. This involves specifying the time duration over which the simulations will be conducted. It is crucial to choose an appropriate simulation time frame that captures the relevant operational scenarios and allows for a comprehensive analysis of the consequences. Then the failure mode scenarios are run and future states of the system are saved for further analysis and evaluation of the consequences, as well as identification of potential mitigation strategies. It is usual to quantify the consequences by combining the probability of occurrences for each failure mode. Researchers and decision-makers can gain valuable insights into the potential impacts of failure modes and make informed decisions regarding risk management, system design improvements, and contingency planning.

The DCA is a common practice for high-risked maritime operations such as Dynamic Positioning (DP) in offshore drilling. Bø, Johansen, Sørensen and Mathiesen (2016) used the DCA to verify the safety of a DP operation in specific to analyze the dynamics of the transient recovery from failure scenarios. They simulated propulsion and power management systems with the loss of a generator set, loss of a thruster, loss of a switchboard, and thruster full thrust failure scenarios. Fossdal (2018) implemented consequence analysis with environmental disturbances and defined failure scenarios to increase situational awareness for autonomous ship navigation. In the study, specific failure modes were selected with predefined probabilities of occurrence. These failure modes include a total power blackout lasting 250 seconds, rudder freeze, and power losses of 80% and 50% for durations of 45 seconds. The DP control algorithm is used for maintaining a defined position and heading for a ship using redundant thrusters and sensors. The redundant propulsion concept as opposed to DP is used for ships in transit. Redundant propulsion emphasizes

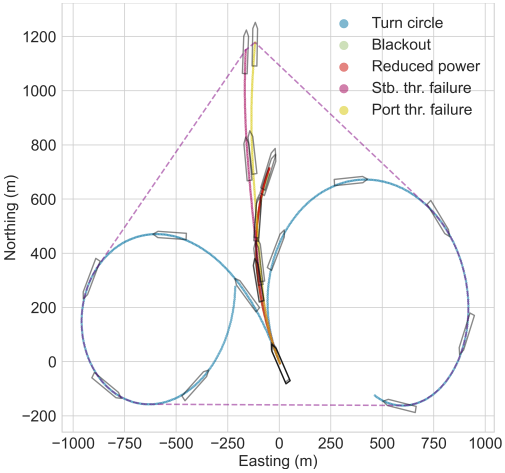


Figure 6: The simulation results of the DCA with 1 m/s ocean current towards 70° and 10 m/s wind towards 50°. Blue, green, red, purple, and yellow lines represent the turn circles, blackout, reduced thruster power, and starboard and port thruster failure scenarios respectively. The safety domain polygon generated from the DCA in this example is presented with a dashed purple line.

redundancy in propulsion systems to ensure operational continuity in the event of failures (DNV, 2022). The failure modes and scenarios considered in this study encompass the following:

i. Turn circles were calculated on both the port and starboard sides by setting the thruster angles to 30°. Simulations were conducted until the turn circles were completed. Turn circle maneuvers are not categorized as failure modes but rather used for considering emergency situations that would require bold maneuvers.

ii. The blackout failure mode scenario was examined, with a simulation time defined as 180 seconds.

iii. The reduced power for thrusters is simulated for 180 seconds. In this scenario, an 80% reduction in power was applied within the thrust allocation matrix.

iv. The thruster failure scenario entailed the deactivation of thrusters one at a time. The ship considered in this study contains two azimuth thrusters for main propulsion and steering. Simulations for this scenario ran for 180 seconds with the starboard azimuth thruster deactivated.

Fig. 6 presents the DCA simulation results with defined failure modes, wind, and ocean currents. Simulation results from the DCA consist of possible positions and heading states of the ship. The DCA function is a computationally demanding operation and takes approximately 600 milliseconds to run in the study with the defined failure modes and simulation times. The trajectory planning algorithm with MOO requires running the DCA several times and is infeasible. To overcome this limitation, the DCA results are used to build the safety domain of the ship. Possible states from the simulation results are converted into a polygon. Safety domain polygon can have various geometric forms depending on the ship's heading and relative speed affected by the environmental forces acting on it.

A computationally efficient method to utilize DCA in optimization is achieved by creating a lookup table in advance using ship speed and heading combinations. Combinations are chosen as discrete ship speeds between minimum navigable and maximum speeds with 1 m/s increments, and heading changes between 0° and 360° with 10° increments. The wind and ocean current information is kept constant during the lookup table creation. Fig. 7 demonstrates an example of using DCA as safety domains along a ship trajectory.

2.9. Trajectory planning with multi-objective optimization

MOO allows considering multiple conflicting or complementary objectives simultaneously while enabling decision-makers to identify solutions that provide a balanced compromise between different performance metrics such as travel time, path length, safety, and fuel consumption. The availability of alternative solutions makes MOO well-suited for deployment in autonomous ship navigation systems, serving as a decision-support tool for remote or

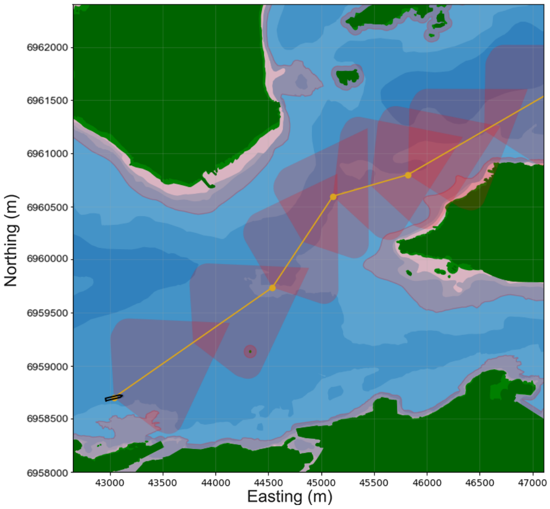


Figure 7: An example of using DCA as safety domains along a ship trajectory.

onboard operators. The MOO problem comprises decision variables, objectives, and constraints. The mathematical representation of a MOO problem is defined as:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), ..., f_k(\boldsymbol{x})]^{\mathrm{T}} \\ \text{subject to} \quad & g_j(\boldsymbol{x}) \leq 0, \qquad j = 1, 2, ..., m \\ & h_l(\boldsymbol{x}) = 0, \qquad l = 1, 2, ..., e \end{aligned} \tag{12}$$

where $\boldsymbol{x}$ is the vector of decision variables, $\boldsymbol{f}(\boldsymbol{x})$ is the vector of objective functions with $f_i(\boldsymbol{x})$ for each objective, $g_i(\boldsymbol{x})$ and $h_i(\boldsymbol{x})$ are inequality and equality constraints.

In MOO, the goal is to find a set of solutions that represents the trade-offs between different objectives, rather than a single optimal solution. These solutions are known as MOO solutions. MOO solutions are typically evaluated based on Pareto dominance. A solution in the feasible solution set ($\boldsymbol{x}^* \in \boldsymbol{X}$) is called Pareto optimal if there is no other solution ($\boldsymbol{x} \in \boldsymbol{X}$) such that $\boldsymbol{f}(\boldsymbol{x}) \leq \boldsymbol{f}(\boldsymbol{x}^*)$, and $f_i(\boldsymbol{x}) < f_i(\boldsymbol{x}^*)$ for at least one objective (Marler and Arora, 2004). The Pareto front refers to the set of non-dominated solutions in the solution space, representing the optimal trade-off between conflicting objectives. In the Pareto front, each solution represents a unique combination of objective values where no other solution can improve on one objective without sacrificing performance in another objective as shown in Fig. 8. The Pareto front provides valuable insights into the feasible solutions that strike a balance between conflicting objectives.

MOO problem definition in this study is to optimize a predefined initial trajectory of autonomous ship considering environmental disturbances, grounding risk, and possible failure scenarios. The initial trajectory consists of waypoints and speed plans between waypoints. In addition to the initial trajectory, it is assumed to have the ENC of the environment as polygons for land and various depths. Ocean current, wind, and tidal information, i.e., water level are also used as inputs in the problem. The water level combined with the dynamic draught of the ship is used in the UKC method explained in section 2.6 to obtain an updated map that contains grounding polygons. In this context, the decision variables pertain to the deviation from initial waypoints within a predefined radius and the deviation from the speed plans in waypoint legs. The defined constraints for the problem include the incorporation of new waypoint legs that adhere to the UKC protocol by avoiding land or shallow waters. Additionally, the course changes between consecutive waypoint legs are limited to a maximum of 60° to prevent abrupt course changes. The MOO objective functions are defined as:

Objective 1: Minimizing safety domain violation. Safety domains are retrieved from the DCA look-up table at each waypoint and waypoint legs, and intersected areas of safety domains (A^i_{dca}) and grounding obstacles

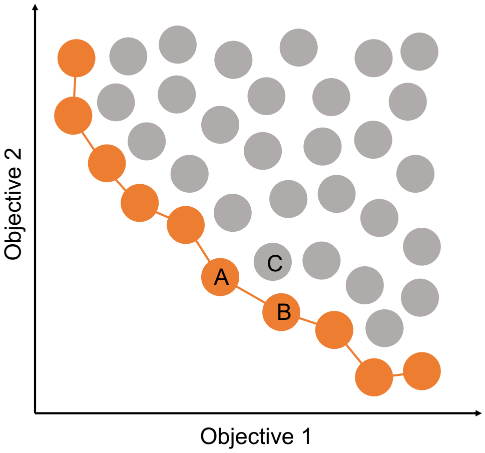


Figure 8: Solutions of a MOO problem with two objectives where orange points represent the Pareto front. Point C is not on the Pareto front since points A and B dominate point C.

($A_{grounding}$) are used to calculate safety domain violation:

$$obj_1 = \frac{1}{n} \sum_{i=1}^{n} \frac{A_{dca}^{i} \cap A_{grounding}}{A_{dca}^{i}} \tag{13}$$

Objective 2: Minimizing speed changes between consecutive waypoint legs, where U_i is the new trajectory's speed plan for the ith waypoint leg:

$$obj_2 = \frac{1}{n} \sum_{i=2}^{n} |U_i - U_{i-1}| \tag{14}$$

Objective 3: Minimizing the new trajectory's length where $wp_i = (x_i, y_i)$ is the position of the ith waypoint:

$$obj_3 = \sum_{i=2}^{n} ||wp_i - wp_{i-1}|| \tag{15}$$

Objective 4: Minimizing deviation from initial waypoints where wp_{init_i} and wp_i are ith waypoints of the initial and new trajectories:

$$obj_4 = \frac{1}{n} \sum_{i=1}^{n} ||wp_i - wp_{init_i}|| \tag{16}$$

Objective 5: Minimizing deviation from the planned ETA. The planned and new ETAs are derived from the initial and new trajectory plans:

$$obj_5 = |ETA_{new} - ETA_{init}| \tag{17a}$$

$$ETA = \sum_{i=2}^{n} \frac{||wp_i - wp_{i-1}||}{U_{i-1}} \tag{17b}$$

There are several algorithms commonly used in MOO, including EA, GA, PSO, and ACO. These algorithms offer advantages such as stochastic nature for global search of optimal solutions, reduced likelihood of being trapped in local optima, suitability for handling complex objectives (nonconvex, mixed discrete/continuous, non-smooth, etc.), and the potential for faster computation through parallelization. For this study, the Multi-objective Particle Swarm Optimization (MOPSO) algorithm has been selected due to its fast convergence speed (Coello, Pulido and Lechuga, 2004). PSO is initially developed by Kennedy and Eberhart (1995) and is inspired by the social behavior of bird flocks and fish schools. In PSO, a population of particles moves through the search space, guided by their individual best solution as well as the best solution found by any particle in the swarm. By continuously adjusting their positions based on these influences, particles tend to converge toward an optimal solution. MOPSO is an extension of the traditional PSO algorithm and is developed by Coello et al. (2004). The key idea behind MOPSO is to enable particles to maintain a set of Pareto-optimal solutions rather than a single best solution. Particles maintain personal archives of non-dominated solutions they have found during their exploration and share their archives with each other, allowing them to learn from the collective experience of the swarm. Maintaining personal archives and sharing information among particles enables MOPSO to converge toward the Pareto front. The following are the steps to implement MOPSO in this study:

Step 1: Initialize a swarm of particles with random deviations from the waypoints and speed plans within defined constraints in the search space.

Step 2: Evaluate the fitness of each particle's position and speed plan based on the multiple objectives being optimized.

Step 3: Each particle updates its individual best position and speed plan and updates its individual archive with non-dominated solutions found so far, representing different trade-offs between the objectives.

Step 4: Particles share information about their archives with neighboring particles in the swarm and adjust their deviation from initial waypoint positions and speed plans to move towards non-dominated solutions that better satisfy the multiple objectives.

Step 5: Update the positions and speed plans of the particles.

Step 6: Repeat steps 2 to 5 for a specific number of iterations or until convergence criteria are met, enabling the swarm to explore the decision space and identify a diverse set of Pareto-optimal solutions.

Step 7: The final individual archives of the particles contain a variety of non-dominated solutions, representing feasible and optimal trade-offs between the deviations from initial waypoints and speed plans.

The MOPSO algorithm from the Platypus framework[1] is utilized to implement the MOO problem in this study.

2.10. Multi-criteria decision-making

While the exploration of the solution space is achieved by the MOPSO that produces alternative solutions, i.e., the Pareto front, the exploitation of the solution space is achieved by MCDM methods. MCDM is about selecting a solution among alternative solutions, or ranking the alternatives that satisfy user-defined criteria. Several commonly employed MCDM techniques include such as the Analytic Hierarchy Process (AHP) (Saaty, 1980), the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) (Tzeng and Huang, 2011), the Multi-Objective Optimization by Ratio Analysis (MOORA)) (Brauers, Ginevičius and Podvezko, 2010), the CRiteria Importance Through Inter-criteria Correlations (CRITIC) (Diakoulaki, Mavrotas and Papayannakis, 1995), and Entropy Weight Method (EWM) (Lotfi and Fallahnejad, 2010).

The ratio system of MOORA is implemented in this study for considering the operator preferences and ranking the Pareto front solution set accordingly. MOORA method utilizes operator insights as user-defined weights for each objective function. The user is requested to assign integer values ranging from 1 (indicating the least importance) to 10 (indicating the highest importance) for each objective. However, the direct allocation of weights brings subjectivity to the approach. AHP can be an alternative method to mitigate subjectivity by asking multiple pairwise questions to operators to understand the preferences more systematically. However, AHP suggests $n(n-1)/2$ pair-wise questions to the user where n is the number of decision criteria or objectives. Instead of asking 10 pair-wise questions to the operator, the MOORA method is chosen in this study considering the time limitations in autonomous navigation operations.

Neither AHP nor MOORA method can prevent subjective weight allocation while methods such as EWM and CRITIC propose objective weight calculation without relying on user inputs. In this study, in addition to the MOORA

[1] The source code is available from https://github.com/Project-Platypus/Platypus.

method, the EWM is applied for automatic weight allocation for objectives which is then used to rank the Pareto front solutions. In addition to the solutions considering user-defined preferences, alternative solutions ranked by the EWM method are proposed to the user. The EWM utilizes the concept of entropy to quantify the diversity or spread of values across multiple criteria, enabling the determination of objective weights and facilitating the ranking of alternatives systematically and comprehensively.

The MCDM problem is expressed in matrix form as:

$$S = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & a_{m4} & a_{m5} \end{bmatrix} \tag{18a}$$

$$w = [w_1, w_2, w_3, w_4, w_5] \tag{18b}$$

Here, S is the decision matrix or the Pareto front solution set with $m \times 5$ dimension where m represents the total number of alternative solutions, 5 is the number of objectives, and a_{mn} are the objective function values calculated by the MOO algorithm. Moreover, w contains weights for each objective and are defined by the operator for the MOORA method and calculated automatically by the EWM. The steps for the ratio system of the MOORA method are explained as:

Step 1: Asking the operator to define the importance of each objective with a value $v_j \in [1, 10]$ where 1 is the least important and 10 is the most important.

Step 2: Normalizing the user-defined values to create a weights vector

$$w_j = \frac{v_j}{\sum_{j=1}^{5} v_j} \tag{19}$$

Step 3: Calculating the normalized decision matrix values with the min-max normalization method

$$x_{ij} = \frac{a_{ij} - min(a_{ij})}{max(a_{ij}) - min(a_{ij})} \tag{20}$$

Step 4: Calculating the weighted normalized decision matrix values

$$W_{ij} = w_j x_{ij} \tag{21}$$

Step 5: Calculating priorities for each alternative solution

$$Q_i = \sum_{j=1}^{n} W_{ij} \tag{22}$$

Step 6: Ranking the alternative solutions using the priority values Q_i

The EWM allows to calculate objective weights automatically without user input and the steps for the EWM are explained as:

Step 1: Calculating the normalized decision matrix values by Eq. (20).

Step 2: Calculating the entropy values for each criterion, i.e., objective

$$e_j = -\frac{\sum_{i=1}^{m} x_{ij} \ln x_{ij}}{\ln m} \tag{23}$$

Step 3: Calculating variation of coefficients

$$g_j = |1 - e_j| \tag{24}$$

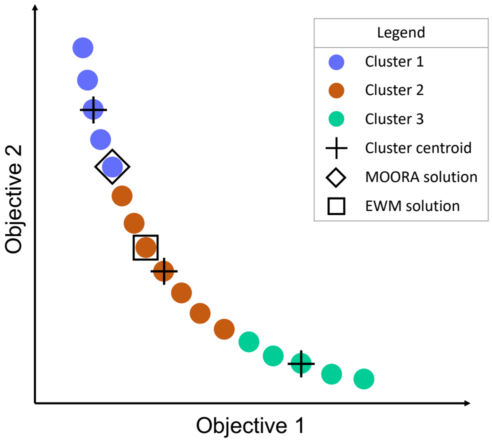


Figure 9: An example of the Pareto front solutions with two objectives. Clusters are represented with different colors, and proposed solutions from clusters, MOORA, and EWM methods are presented with cross, diamond, and square symbols respectively

Step 4: Calculating the weights for the objectives

$$w_j = \frac{g_j}{\sum_{j=1}^{m} g_j} \tag{25}$$

Step 5: Calculating the weighted normalized decision matrix values by the Eq. (21)

Step 6: Calculating priorities for each alternative solution by the Eq. (22)

Step 7: Ranking the alternative solutions using the priority values.

Both the MOORA and EWM methods are capable of ranking the solutions on the Pareto front. However, depending on the problem, MOO can generate an overwhelming number of solutions. To reduce the workload of operators, a specified number of solutions can be presented from the ranked lists. Furthermore, it is a common approach to prune the Pareto front solutions and only present the representative solutions to the decision-maker (Petchrompo, Coit, Brintrup, Wannakrairot and Parlikad, 2022). A diversity-based clustering method is applied in this study for pruning the Pareto front solutions. To achieve this, an unsupervised machine learning algorithm, i.e., the K-means clustering algorithm (Hartigan and Wong, 1979), is used for grouping the solutions. The algorithm partitions data points into a predetermined number of clusters, each represented by a central point known as a centroid. Through an iterative process, K-means aims to minimize the sum of squared distances between data points and their respective centroids, effectively grouping similar data points together. The solutions that are closest to the centroid of each cluster are considered representative solutions. Before using the K-means algorithm, it is necessary to know the total number of clusters beforehand. Since the number of clusters changes according to the individual problem scenario, it is not proper to define a constant value. Therefore, the silhouette cluster analysis method (Rousseeuw, 1987) is applied to objectively determine the optimal number of clusters before the K-means clustering. The silhouette method quantifies the quality of clustering by evaluating intra-cluster and nearest-cluster distances. Fig. 9 presents an example of the Pareto front solutions with two objectives. The Pareto front is grouped into three clusters. Assuming the user prefers objective 1 over objective 2, the MOORA method finds the solution marked with the diamond symbol, while the EWM method proposes the solution with the square symbol. Cross symbols represent the solutions closest to the cluster centroids. After all, the algorithm proposes five alternative solutions to the user for this example.

3. Results

3.1. Scenario 1

For the first scenario, a dangerous initial trajectory is chosen that passes between an island and the mainland. Constant ocean current with 0.5 m/s towards 070 and wind with 10 m/s towards 050 is defined. User preferences are defined as $\boldsymbol{v} = [10, 4, 3, 2, 5]$ for the DCA violation, consecutive speed changes, path length, deviation from the initial trajectory, and deviation from the ETA respectively. Then, user-defined preferences are converted to normalized weights $\boldsymbol{w}_{moora} = [0.416, 0.166, 0.125, 0.083, 0.208]$ to be used in the MOORA method. Additionally, the EWM method calculated weights as $\boldsymbol{w}_{entropy} = [0.20042, 0.19990, 0.20002, 0.19983, 0.19981]$. The EWM weights are utilized to compute the weighted summation feature, which serves to prioritize objectives within groups and propose the top-ranked solutions as alternatives to the user. Fig. 10 (a) visualizes all the alternative solutions found by the MOPSO algorithm, (b) the top-ranked solution calculated by the MOORA method considering user-defined preferences, (c) the top-ranked solution calculated by the EWM method, (d)-(e) representative solutions of the two clusters calculated by silhouette and K-means clustering methods, (f) Simulation result presenting the final ship trajectory for a chosen solution, and (g) presents the Pareto front graph with clusters and proposed solutions. The MOPSO algorithm finds 58 alternative solutions in 36 seconds and Table 1 presents objective function values for the proposed alternative solutions both before and after normalization. Additionally, compared to the initial trajectory, new trajectory improvements for objective functions are demonstrated in percentage. With $\boldsymbol{w}_{moora}$, the user gave the highest preference to the DCA violation, so the MOORA method ranked the solution with the lowest DCA violation among other alternatives. However, this resulted in the longest path length. The EWM calculated a balanced weight vector and this method ranked a solution with the second lowest DCA violation and lowest values for consecutive speed changes, deviation from the initial trajectory, and ETA. Two cluster solutions had similar objective values and geometry, however, the second cluster has a shorter path length than the first cluster.

3.2. Scenario 2

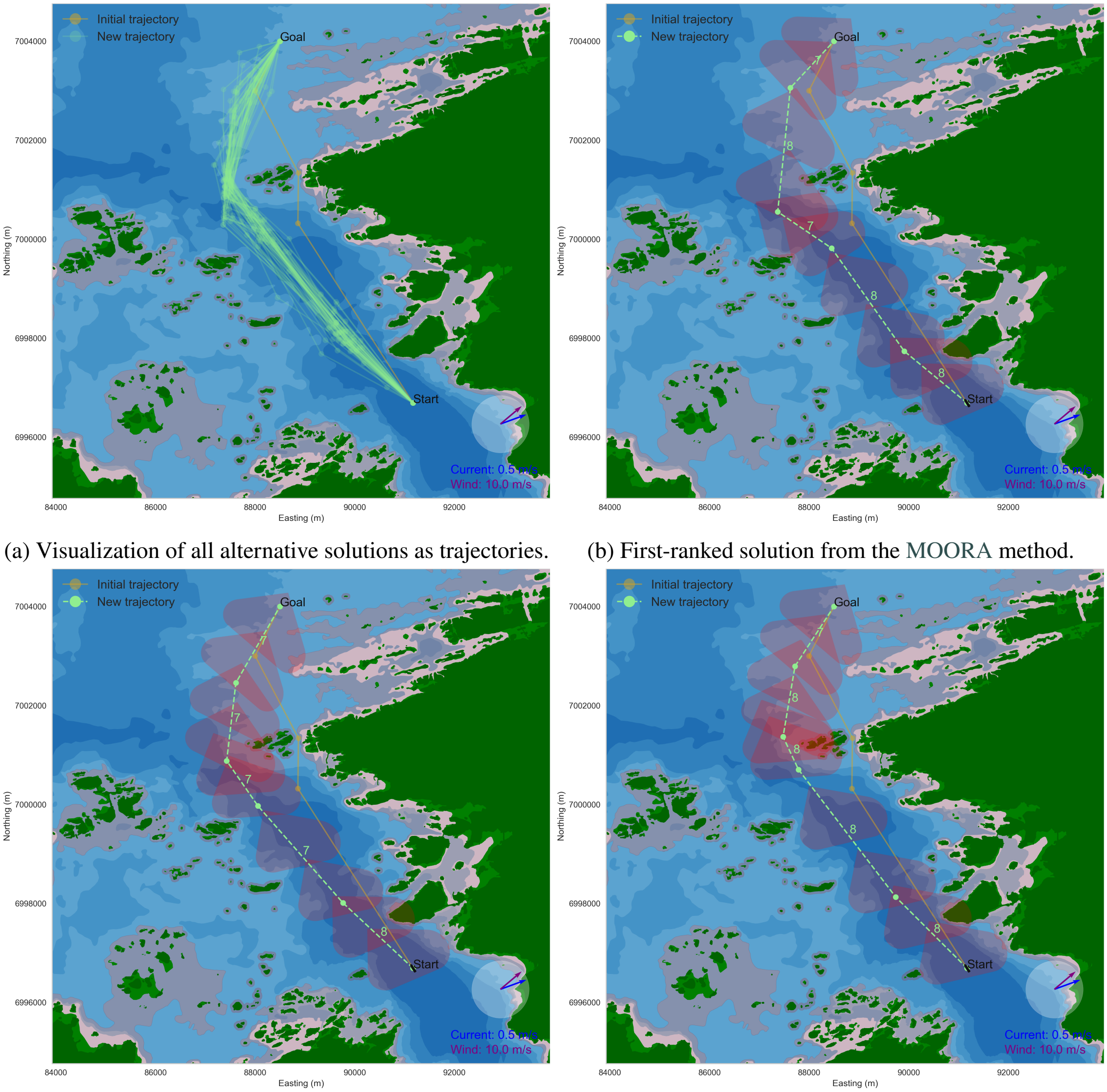
In this scenario, the algorithm's performance to mitigate grounding risk caused by environmental disturbances is demonstrated. Constant ocean current with 0.5 m/s towards 350 and wind with 10 m/s towards 0 is defined to propose grounding risk towards the land on the top of the figures. The same user preferences are used as in scenario 1. The MOPSO algorithm finds 116 alternative solutions in 34 seconds. First-ranked solutions from the MOORA and EWM methods are combined with two cluster-representative solutions. This combined set of solutions is then presented to the operator for the final decision-making process, as illustrated in Fig. 11. Table 1 presents the objective function values for the proposed alternative solutions. As expected, the algorithm suggests new trajectories taking into account the current and wind directions, in order to move away from the shore. The new trajectories do not result in longer distances, therefore the proposed speed plans are now closer to the initial speed plan, which remains constant at 8 m/s. The MOORA method solution has the smallest DCA violation value among the others because of the defined user preferences. To address a smaller DCA violation value, the algorithm computes a new trajectory that passes through the middle of the narrow channel which would cause a problem if there is another vessel in the area. The first-ranked solution of the EWM method presents a good trade-off between safety domain violation, consecutive speed changes, and deviation from the ETA, and is selected as the final trajectory for the simulation.

3.3. Scenario 3

Scenario 3 aims to demonstrate the UKC protocol's effect on trajectory planning. The scenario takes place in a narrow channel. Constant ocean current with 0.5 m/s towards 070 and wind with 10 m/s towards 050 is defined and the same user preferences are used as in scenarios 1 and 2. The scenario is run two times with different UKC protocols to define different shallow water contours. Fig. 12 (a) demonstrates the proposed solutions when following a UKC protocol (C_{ukc_1}) that requires minimum depths equivalent to half of the ship's draught. The proposed solutions are the combined set of top-ranked MOORA and EWM solutions and two cluster representatives. The speed plan is not shown in the figure to reduce visual complexity. Fig. 12 (b) visualizes the third cluster representative solution. And Fig. 12 (c) and (d) demonstrate the proposed solutions when the UKC protocol requires minimum depths equivalent to double the ship's draught (C_{ukc_2}). Table 1 lists the objective function results for the solutions. The algorithm finds 121 alternative solutions in 34 seconds for the scenario for $\boldsymbol{C}_{ukc_1}$ and 69 alternative solutions in 32 seconds for $\boldsymbol{C}_{ukc_2}$. It is seen in Table 1 that the DCA violation values of $\boldsymbol{C}_{ukc_2}$ increased due to the reduced navigational waters.

4. Discussion

In this section, the information obtained from the study will be utilized to address the research questions outlined in the introduction.



(a) Visualization of all alternative solutions as trajectories.

(b) First-ranked solution from the MOORA method.

(c) First alternative solution from the EWM method.

(d) Representative solution of the first cluster (C1).

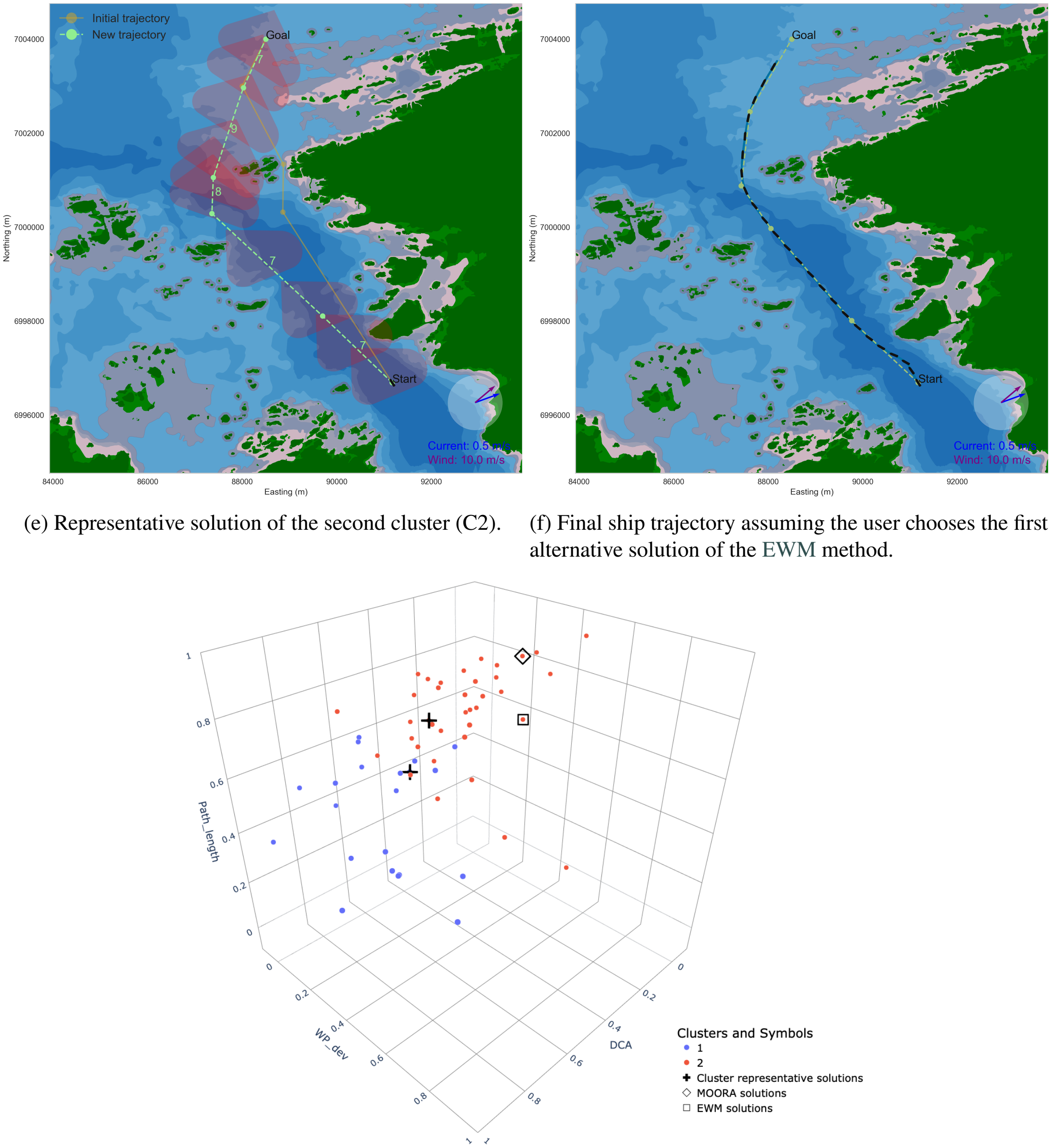
Figure 10: Trajectories calculated by the MOPSO algorithm for Scenario 1.

Question 1: In what ways can autonomous ship navigation be enhanced through the integration of environmental forces, navigable waters, maneuvering characteristics, and potential failure modes?

When planning the trajectory of an autonomous ship, it is important to take into account factors beyond just chart depths. The actual depths of water can vary depending on the tide and weather conditions. High speeds in narrow channels can cause a squat effect and lead to the ship running aground in shallow areas. The ship's maneuvering abilities, such as stopping distance and turning circle, are also influenced by its speed. To ensure safety, risk assessment methods like DCA can be used to anticipate potential failure scenarios and incorporate maneuvering characteristics into the safety domain creation.

Question 2: What advantages and obstacles are associated with the utilization of a decision support system based on MOO for autonomous ship navigation?

One advantage is that a decision support system based on MOO can treat multiple conflicting objectives independently rather than merging them into a single objective. Moreover, this method creates numerous alternate



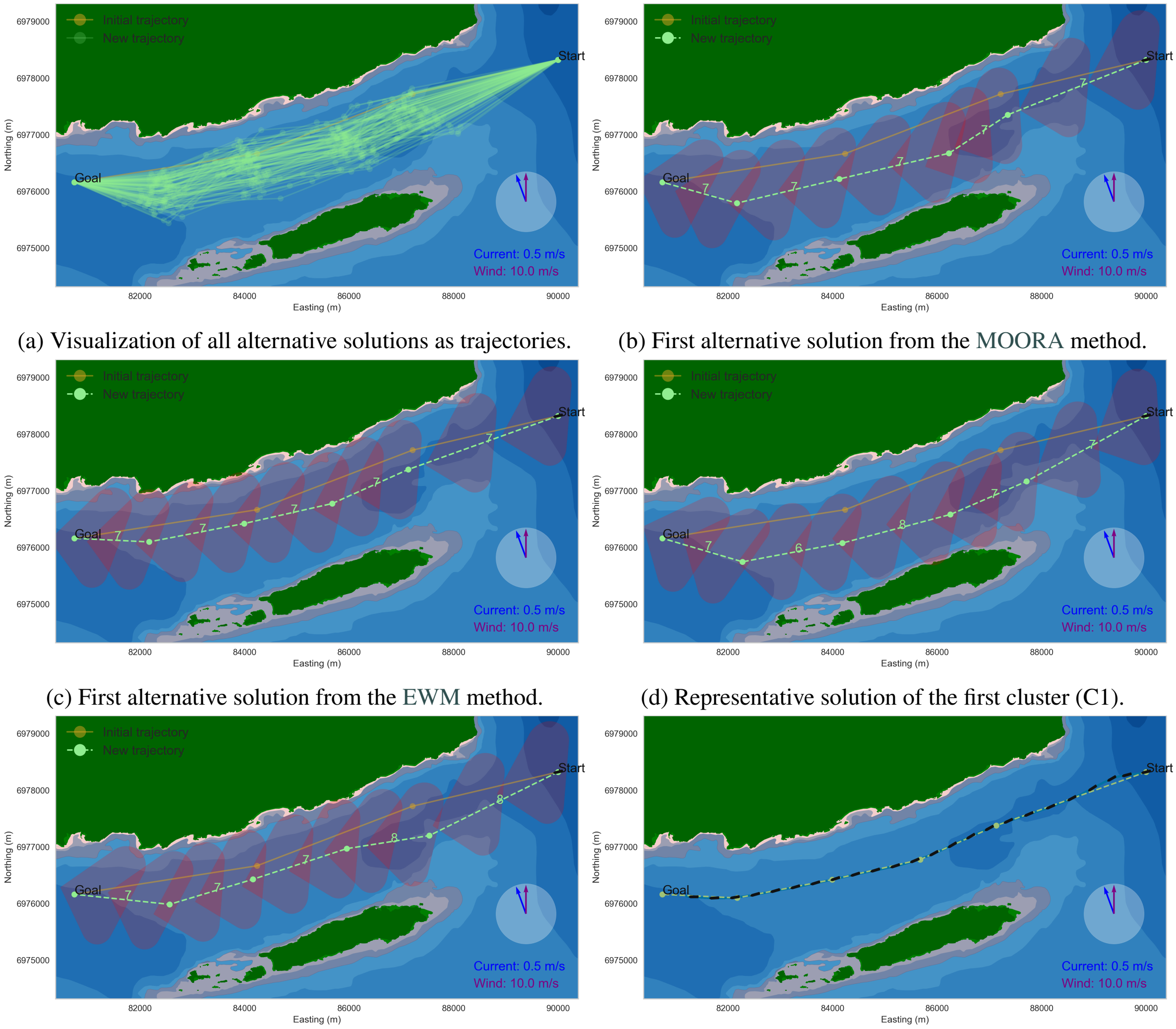
(e) Representative solution of the second cluster (C2).

(f) Final ship trajectory assuming the user chooses the first alternative solution of the EWM method.

(g) The Pareto front solutions with clusters and proposed solutions.

Figure 10: Trajectories calculated by the MOPSO algorithm for Scenario 1.

solutions, giving users a range of choices. A drawback is, solving the MOO problem can take time depending on the problem definition, algorithm, hardware system, and especially if the dynamic obstacles' future trajectories are considered. Table 2 is prepared to present the study's computational efficiency. The table presents descriptive statistics of scenarios' execution times in seconds. Each scenario is run 10 times on a PC with ARM M1 8 Core 3200 MHz processor and 16 GB of LPDDR4 RAM. The primary determinant influencing computational time is the dimensionality of the decision variables. Results from scenarios with 4 waypoints to optimize exhibited comparable execution times. Notably, as the dimensionality of the decision variables, specifically the number of waypoints to optimize, increased, so did the corresponding execution time.



(a) Visualization of all alternative solutions as trajectories.

(b) First alternative solution from the MOORA method.

(c) First alternative solution from the EWM method.

(d) Representative solution of the first cluster (C1).

(e) Representative solution of the second cluster (C2).

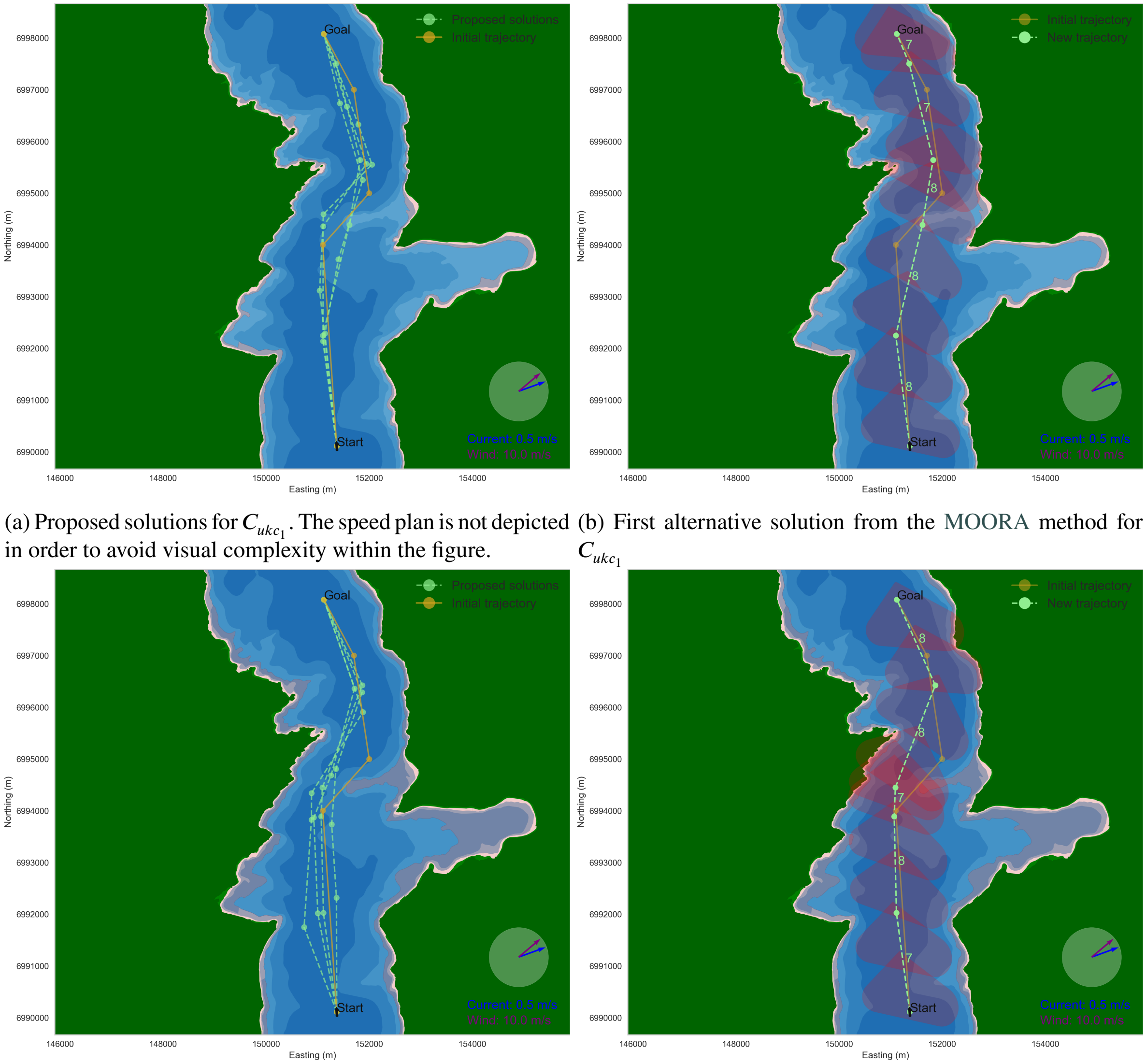
(f) Final ship trajectory assuming the user chooses the first alternative solution of the EWM method.

Figure 11: Trajectories calculated by the MOPSO algorithm for Scenario 2.

In the simulations, the algorithm generated numerous alternative solutions. Therefore, it is important to continue investigating alternative methods for evaluating, eliminating, and presenting only a few alternative solutions to the operators. Specific to this method, the stochastic nature of the MOPSO algorithm leads to varied outcomes based on the initializations of particles. This variability can occasionally result in convergence to local minima. To address this challenge, employing strategies such as performing multiple runs with distinct initializations, extending the number of iterations, and optimizing parameters to emphasize exploration over exploitation can be advantageous. These approaches can enhance the algorithm's capacity to explore the solution space and avoid premature convergence, however, requires increased computational effort.

Question 3: What strategies can be employed to foster collaboration between autonomous systems and operators during the decision-making process of ship navigation?

While autonomous systems excel in speed, accuracy, and multitasking, human operators possess unique qualities such as adaptability, emotional intelligence, creativity, and ethical reasoning. Evaluating the strengths and weaknesses of each information processing capability is essential for designing effective human-machine collaboration. This study aims for human-machine collaboration by utilizing MOO algorithms to find multiple alternative solutions meeting the



(a) Proposed solutions for C_{ukc_1}. The speed plan is not depicted in order to avoid visual complexity within the figure.

(b) First alternative solution from the MOORA method for C_{ukc_1}

(c) Proposed solutions for C_{ukc_2}. The speed plan is not depicted in order to avoid visual complexity within the figure.

(d) Representative solution of the first cluster (C1) for C_{ukc_2}

Figure 12: Trajectories calculated by the MOPSO algorithm for Scenario 3.

objective criteria, and ranking the solutions by using the human operator preferences for each objective. Additionally, the Pareto front solutions are ranked by automatic weighting with the entropy method and cluster representative solutions are retrieved to prevent subjectivity of the human preferences. Depending on the problem definition, it is experienced in this study that some solutions came up high in the ranking although they are not preferable by the authors. This serves as a valuable reminder to not solely depend on solutions suggested by autonomous systems, but to also involve human evaluation.

5. Conclusion

This study focuses on human-machine collaboration through the development of a decision support system for autonomous ship trajectory planning. The system considers environmental forces like ocean currents, wind, and tidal information, while also accounting for narrow channel geometry, squat effect, and UKC. Utilizing a 3-DOF ship maneuvering model, a DCA risk assessment method is employed to establish dynamic safety domains by incorporating ship maneuvering characteristics and potential failure scenarios. Leveraging these features, the MOPSO algorithm is implemented to generate alternative trajectories aiming to minimize five objective functions, encompassing safety domain violation, consecutive speed change, path length, deviation from the initial plan, and deviation from the initial ETA. Additionally, two MCDM methods, MOORA with manual user preferences input and EWM with automatic weight allocation are utilized to rank alternative solutions from the Pareto front. Ultimately, the Pareto front solutions are clustered with the K-means algorithm, leading to the selection of representative solutions from each cluster centroid. These cluster representatives are then merged with the highest-ranked alternative solutions determined by MCDM methods. The combination of these solutions guides the final decision-making process carried out by the operator. Notably, the study does not consider dynamic obstacles, and it was evaluated through three specific scenarios.

Prior to concluding, let us consider some potential areas for future work. Exploring alternative MOO algorithms, such as NSGAII, NSAGIII, ACO, etc., within the context of this study structure would be of significant interest. Similarly, other MCDM techniques, such as AHP, TOPSIS, etc., can be employed to explore the ranking of solutions on the Pareto front. Furthermore, other clustering methods can be utilized to represent a smaller number of solutions from the Pareto front. Lastly, this study considers the grounding risk while does not contain a collision avoidance method for dynamic obstacles. The collision avoidance algorithms for dynamic obstacles work for smaller spatial and temporal domains. The decision support system proposed in this study covers grounding risk and requires a slower time scale because of the nature of the human decision-making process. The method can be used during the planning phase before an autonomous ship mission or it can run periodically during the trajectory following phase while collaborating with the human operators. Nevertheless, implementing a collision avoidance trajectory planning algorithm with dynamic obstacles using a hierarchical approach could yield valuable insights. The new trajectory plans considering grounding risk, environmental forces, and DCA from this method can be input into a collision avoidance algorithm that addresses collision risk and the COLREG rules.

6. Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used ChatGPT 3.5 and Grammarly in order to improve the readability and language of the work. After using these tools/services, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Table 1
The MOO objective function results for scenarios. Objective function results are presented as values after and before normalization and respective solution's improvement in percentage.

Scenario 1		DCA			Speed changes			Path length			Deviation from initial WPs			Deviation from ETA		
Index	Source	Norm.	Org.	Impr.	Norm.	Org.	Impr.	Norm.	Org.	Impr.	Norm.	Org.	Impr.	Norm.	Org.	Impr.
1	MOORA	0.000	0.020	-%76	0.272	0.600	N/A	0.828	10513.7	%12	0.230	877.47	N/A	0.057	19.47	-%13
2	EWM	0.039	0.022	-%72	0.272	0.600	N/A	0.620	9823.85	%8	0.280	919.29	N/A	0.09	33.29	-%9
3	C1	0.513	0.046	-%59	0.363	0.800	N/A	0.559	9616.41	%7	0.205	855.94	N/A	0.478	160.30	-%18
4	C2	0.461	0.043	-%63	0.363	0.800	N/A	0.737	10213.41	%10	0.248	892.42	N/A	0.013	4.62	-%11

Scenario 2		DCA			Speed changes			Path length			Deviation from initial WPs			Deviation from ETA		
Index	Source	Norm.	Org.	Impr.	Norm.	Org.	Impr.	Norm.	Org.	Impr.	Norm.	Org.	Impr.	Norm.	Org.	Impr.
1	MOORA	0.011	0.003	-%97	0.111	0.0200	N/A	0.378	8198.22	%2	0.396	605.99	N/A	0.073	20.86	%17
2	EWM	0.154	0.030	-%75	0.000	0.000	N/A	0.480	8524.54	%1	0.219	435.35	N/A	0.091	25.75	%15
3	C1	0.066	0.013	-%92	0.555	1.000	N/A	0.288	7910.84	%3	0.516	721.63	N/A	0.191	53.78	%18
4	C2	0.100	0.019	-%88	0.222	0.400	N/A	0.539	8711.66	%2	0.250	465.01	N/A	0.065	18.41	%9

Scenario 3 for C_{ukc_1}		DCA			Speed changes			Path length			Deviation from initial WPs			Deviation from ETA		
Index	Source	Norm.	Org.	Impr.	Norm.	Org.	Impr.	Norm.	Org.	Impr.	Norm.	Org.	Impr.	Norm.	Org.	Impr.
1	MOORA	0.022	0.003	-%88	0.052	0.200	N/A	0.644	7962.50	-%4	0.256	462.20	N/A	0.071	23.55	%1
2	EWM	0.058	0.006	-%87	0.105	0.400	N/A	0.606	7829.20	-%4	0.130	309.40	N/A	0.044	14.50	%6
3	C1	0.127	0.012	-%72	0.105	0.400	N/A	0.576	7722.79	-%2	0.183	372.83	N/A	0.111	36.63	%7
4	C2	0.317	0.030	-%25	0.210	0.800	N/A	0.308	6787.57	%1	0.426	667.81	N/A	0.327	107.11	%20

Scenario 3 for C_{ukc_2}		DCA			Speed changes			Path length			Deviation from initial WPs			Deviation from ETA		
Index	Source	Norm.	Org.	Impr.	Norm.	Org.	Impr.	Norm.	Org.	Impr.	Norm.	Org.	Impr.	Norm.	Org.	Impr.
1	MOORA	0.022	0.003	-%88	0.052	0.200	N/A	0.644	7962.50	-%4	0.256	462.20	N/A	0.071	23.55	%1
1	MOORA	0.113	0.031	-%60	0.076	0.200	N/A	0.424	7365.59	-%3	0.233	509.55	N/A	0.028	9.49	%11
2	EWM	0.132	0.034	-%52	0.076	0.200	N/A	0.388	7217.80	-%4	0.152	443.33	N/A	0.092	30.60	%10
3	C1	0.236	0.048	-%39	0.230	0.600	N/A	0.503	7693.14	-%2	0.103	402.35	N/A	0.167	55.63	%2
4	C2	0.207	0.044	-%36	0.384	1.000	N/A	0.256	6672.73	%0	0.674	870.77	N/A	0.584	193.80	%5

Table 2
Simulation execution times for each scenario in seconds.

Scenario	Count	Minimum	Median	Mean	Maximum	Std. Dev.
1	10	35	36	36.4	39	1.17
2	10	30	31	31.6	34	1.42
3	10	31	31.5	31.8	34	1.03
Total	30	30	32	33.2	39	2.5

References

Barrass, B., Derrett, C.D., 2011. Ship stability for masters and mates. Elsevier.

Blendermann, W., 1994. Parameter identification of wind loads on ships. Journal of Wind Engineering and Industrial Aerodynamics 51, 339–351.

Blindheim, S., Johansen, T.A., 2021. Electronic navigational charts for visualization, simulation, and autonomous ship control. IEEE Access 10, 3716–3737.

Blindheim, S., Johansen, T.A., 2022. Particle swarm optimization for dynamic risk-aware path following for autonomous ships. IFAC-PapersOnLine 55, 70–77.

Blindheim, S., Rokseth, B., Johansen, T.A., 2023. Autonomous machinery management for supervisory risk control using particle swarm optimization. Journal of Marine Science and Engineering 11, 327.

Bø, T.I., Johansen, T.A., Sørensen, A.J., Mathiesen, E., 2016. Dynamic consequence analysis of marine electric power plant in dynamic positioning. Applied Ocean Research 57, 30–39.

Brauers, W.K.M., Ginevičius, R., Podvezko, V., 2010. Regional development in lithuania considering multiple objectives by the moora method. Technological and economic development of economy 16, 613–640.

Burmeister, H.C., Constapel, M., 2021. Autonomous collision avoidance at sea: A survey. Frontiers in Robotics and AI 8, 739013.

Cockcroft, A.N., Lameijer, J.N.F., 2003. Guide to the collision avoidance rules. Elsevier.

Coello, C.A.C., Pulido, G.T., Lechuga, M.S., 2004. Handling multiple objectives with particle swarm optimization. IEEE Transactions on evolutionary computation 8, 256–279.

Diakoulaki, D., Mavrotas, G., Papayannakis, L., 1995. Determining objective weights in multiple criteria problems: The critic method. Computers & Operations Research 22, 763–770.

DNV, 2022. Part 6 additional class notations, chapter 2 propulsion, power generation and auxiliary systems, section 7 redundant and alternative propulsion. Rules for Classification: Ships, Edition July 2022 .

Fossdal, M., 2018. Online Consequence Analysis of Situational Awareness for Autonomous Vehicles. Master's thesis. NTNU.

Fossen, T.I., 2021. Handbook of marine craft hydrodynamics and motion control. 2nd ed., John Wiley & Sons.

Fossen, T.I., 2023. An adaptive line-of-sight (alos) guidance law for path following of aircraft and marine craft. IEEE Transactions on Control Systems Technology .

Gillies, S., van der Wel, C., Van den Bossche, J., Taves, M.W., Arnott, J., Ward, B.C., et al., 2023. Shapely. URL: `https://github.com/shapely/shapely`, doi:`10.5281/zenodo.5597138`.

Hartigan, J.A., Wong, M.A., 1979. Algorithm as 136: A k-means clustering algorithm. Journal of the royal statistical society. series c (applied statistics) 28, 100–108.

Hu, L., Naeem, W., Rajabally, E., Watson, G., Mills, T., Bhuiyan, Z., Raeburn, C., Salter, I., Pekcan, C., 2019. A multiobjective optimization approach for colregs-compliant path planning of autonomous surface vehicles verified on networked bridge simulators. IEEE Transactions on Intelligent Transportation Systems 21, 1167–1179.

Huang, Y., Chen, L., Chen, P., Negenborn, R.R., Van Gelder, P., 2020. Ship collision avoidance methods: State-of-the-art. Safety science 121, 451–473.

IMO, 1972. International regulations for preventing collisions at sea (COLREGS).

IMO, 2021. Outcome of the regulatory scoping exercise for the use of maritime autonomous surface ships (MASS). `https://wwwcdn.imo.org/localresources/en/MediaCentre/PressBriefings/Documents/MSC.1-Circ.1638%20-%20Outcome%20Of%20The%20Regulatory%20Scoping%20ExerciseFor%20The%20Use%20Of%20Maritime%20Autonomous%20Surface%20Ships...%20(Secretariat).pdf` (accessed 28.07.2023).

Kennedy, J., Eberhart, R., 1995. Particle swarm optimization, in: Proceedings of ICNN'95-international conference on neural networks, IEEE. pp. 1942–1948.

Lazarowska, A., 2017a. Multi-criteria aco-based algorithm for ship's trajectory planning. TransNav: International Journal on Marine Navigation and Safety of Sea Transportation 11.

Lazarowska, A., 2017b. A new deterministic approach in a decision support system for ship's trajectory planning. Expert Systems with Applications 71, 469–478.

Lotfi, F.H., Fallahnejad, R., 2010. Imprecise shannon's entropy and multi attribute decision making. Entropy 12, 53–62.

Marler, R.T., Arora, J.S., 2004. Survey of multi-objective optimization methods for engineering. Structural and multidisciplinary optimization 26, 369–395.

Negenborn, R.R., Goerlandt, F., Johansen, T.A., Slaets, P., Banda, O.A.V., Vanelslander, T., Ventikos, N.P., 2023. Autonomous ships are on the horizon: here's what we need to know. Nature 615, 30–33.

Ngatchou, P., Zarei, A., El-Sharkawi, A., 2005. Pareto multi objective optimization, in: Proceedings of the 13th international conference on, intelligent systems application to power systems, IEEE. pp. 84–91.

Öztürk, Ü., Akdağ, M., Ayabakan, T., 2022. A review of path planning algorithms in maritime autonomous surface ships: Navigation safety perspective. Ocean Engineering 251, 111010.

Petchrompo, S., Coit, D.W., Brintrup, A., Wannakrairot, A., Parlikad, A.K., 2022. A review of pareto pruning methods for multi-objective optimization. Computers & Industrial Engineering 167, 108022.

Rokseth, B., Haugen, O.I., Utne, I.B., 2019. Safety verification for autonomous ships, in: MATEC web of conferences, EDP Sciences. p. 02002.

Rousseeuw, P.J., 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics 20, 53–65.

Saaty, T.L., 1980. The analytic hierarchy process: planning, priority setting, resource allocation. (No Title) .

Szlapczynski, R., Szlapczynska, J., 2012. On evolutionary computing in multi-ship trajectory planning. Applied Intelligence 37, 155–174.

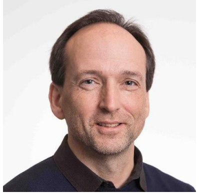
Thieme, C.A., Rokseth, B., Utne, I.B., 2023. Risk-informed control systems for improved operational performance and decision-making. Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability 237, 332–354.

Tzeng, G.H., Huang, J.J., 2011. Multiple attribute decision making: methods and applications. CRC press.

Vagale, A., Oucheikh, R., Bye, R.T., Osen, O.L., Fossen, T.I., 2021. Path planning and collision avoidance for autonomous surface vehicles i: a review. Journal of Marine Science and Technology , 1–15.

Melih Akdag is a Ph.D. candidate in engineering cybernetics at the Norwegian University of Science and Technology (NTNU), Trondheim, Norway. He received his BS in electric/electronic engineering from the Turkish Naval Academy in 2008 and his M.Sc. in marine and coastal protection program from Istanbul University in 2019. After serving as a Navy Officer and Navy Diver in the Turkish Navy with several years of field experience, he is now conducting his research on collaborative collision avoidance algorithms for autonomous ships. He is affiliated with the projects Centre for Autonomous Marine Operations and Systems (AMOS) and the Center for Research-based Innovation SFI AutoShip.

Tom Arne Pedersen has an MSc (2002) and Ph.D. (2009) in Marine Technology from the Norwegian University of Science and Technology (NTNU). He has worked for Marine Cybernetics since 2008, holding positions as a senior project engineer, R&D Manager Drilling Systems, and Product Manager Drilling Systems. Marine Cybernetics was acquired by DNV in 2014 and Tom Arne Pedersen currently holds the position of Principle Researcher in DNV Group Research and Development, working on a testing framework for automated collision and grounding avoidance systems.



Thor I. Fossen is a professor of guidance, navigation, and control (GNC) at the Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim. He received an M.Sc. in Marine Technology in 1987 and a Ph.D. in Engineering Cybernetics in 1991. Besides cybernetics, Fossen's field of research is aerospace engineering and marine technology. This includes GNC systems for uncrewed vehicles (AUV, UAV, USV), robotics, vehicle dynamics, and inertial navigation systems. He has been a Fulbright scholar in flight control at the Department of Aeronautics and Astronautics of the University of Washington, Seattle. Fossen is the author of the Wiley textbook Handbook of Marine Craft Hydrodynamics and Motion Control (2021). He is one of the co-founders and former Vice President of R&D of the company Marine Cybernetics, which DNV acquired in 2012. He is also a co-founder of ScoutDI (2017), which develops drone-based systems for fully digitalized inspections of industrial confined spaces. The Institute of Electrical and Electronics Engineers elevated him to IEEE Fellow in 2016. He received the Automatica Prize Paper Award in 2002 and the Arch T. Colwell Merit Award in 2008 at the SAE World Congress. He was elected to the Norwegian Academy of Technological Sciences (1998) and the Norwegian Academy of Science and Letters (2022).

Tor Arne Johansen received the M.Sc. and Ph.D. degrees in engineering cybernetics from the Norwegian University of Science and Technology (NTNU), in 1989 and 1994, respectively. He is currently a Professor at the Department of Engineering Cybernetics, NTNU. He is also a Key Scientist at the Centre for Autonomous Marine Operations and Systems (AMOS), and the Centre for Research-based Innovation SFI AutoShip, NTNU. He has published several hundred articles in the areas of control, estimation and optimization with applications in the marine, aerospace, automotive, biomedical and process industries. In 2002 Johansen co-founded the company Marine Cybernetics AS where he was Vice President until 2008. Prof. Johansen received the 2006 Arch T. Colwell Merit Award of the SAE. He is the director of the Unmanned Aerial Vehicle Laboratory at NTNU and the SmallSat Laboratory at NTNU. He recently co-founded the spin-off companies ScoutDI, UBIQ Aerospace, Zeabuz, and SentiSystems.