


Article

COLREG-Compliant Optimal Path Planning for Real-Time Guidance and Control of Autonomous Ships

Raphael Zaccone 

DITEN—Department of Naval Architecture, Electrical, Electronic and Telecommunications Engineering, Polytechnic School, University of Genoa, 16145 Genoa, Italy; raphael.zaccone@unige.it

Abstract: While collisions and groundings still represent the most important source of accidents involving ships, autonomous vessels are a central topic in current research. When dealing with autonomous ships, collision avoidance and compliance with COLREG regulations are major vital points. However, most state-of-the-art literature focuses on offline path optimisation while neglecting many crucial aspects of dealing with real-time applications on vessels. In the framework of the proposed motion-planning, navigation and control architecture, this paper mainly focused on optimal path planning for marine vessels in the perspective of real-time applications. An RRT*-based optimal path-planning algorithm was proposed, and collision avoidance, compliance with COLREG regulations, path feasibility and optimality were discussed in detail. The proposed approach was then implemented and integrated with a guidance and control system. Tests on a high-fidelity simulation platform were carried out to assess the potential benefits brought to autonomous navigation. The tests featured real-time simulation, restricted and open-water navigation and dynamic scenarios with both moving and fixed obstacles.



Citation: Zaccone, R. COLREG-Compliant Optimal Path Planning for Real-Time Guidance and Control of Autonomous Ships. *J. Mar. Sci. Eng.* **2021**, *9*, 405. <https://doi.org/10.3390/jmse9040405>

Academic Editor: Salman Nazir

Received: 9 March 2021

Accepted: 8 April 2021

Published: 11 April 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: autonomous ship; autonomous navigation; collision avoidance; COLREG compliance; path planning; path optimisation; guidance and control; ship simulation; RRT*

1. Introduction

Autonomous ships play one of the primary roles in the current research world. While engineers deal with autonomous navigation and control, lawyers and insurance companies try to settle the infinite controversies that the supposed existence of a ship without humans on-board can generate. Meanwhile, as reported by the European Maritime Safety Agency in the annual report [1], the total number of marine accidents is continuously increasing, and collisions and groundings still represent the major accident factors for a ship, as they represent 31% of the total. Despite the high technological development of detection systems, the presence of RADAR (Radio Detection And Ranging) and AIS (Automatic Identification System) on-board ships and the fact that COLREG rules [2] include all the prescriptions for collision prevention, environmental and human factors still play their role to cause the accidents: poor visibility, caused by fog or darkness, rough seas or wind and the tiredness, stress and carelessness of the bridge operators coincide to produce severe consequences. It is not surprising that collision avoidance is the main topic when dealing with ships' autonomous navigation.

Autonomous navigation of marine vessels is a broad topic. Multiple autonomy levels can be recognised: The adoption of decision support systems, aiming to suggest actions to human operators, such as the rudder angle or course changes to avoid collisions [3,4], represents, for instance, a low level of autonomy. In contrast, a ship capable of actuating evasive manoeuvres without human intervention achieves a high autonomy level. The computation of evasive manoeuvres has been addressed to a certain degree for some decades [5]. The scientific literature includes a wide variety of methods, for instance genetic algorithms [6–8] and potential field methods [9]. Tam et al. [10] reviewed the

most common collision avoidance approaches in the maritime field. It is essential to notice that several published studies neglect many crucial issues related to real-world applications of the proposed methods, such as obstacle detection, motion control and uncertainties. As stated by Panda et al. [11], presenting a review of the path-planning methods adopted in the field of autonomous underwater vehicles, autonomous route planning can deal either with the predictable and unpredictable environment: while the first case requires higher level planning, based on available information, such as bathymetry or weather forecasts, the second needs to react to unexpected changes in the surrounding environment. Path planning in a predictable environment includes, for instance, offline weather routing methods and decision support systems [12,13] and can rely on computationally heavy optimisation techniques, as computation time is less impactful. Fully autonomous navigation, on the contrary, requires reactive online strategies that can deal with unexpected changes: obstacle detection, situational awareness and computation time, and efficient guidance and control become matters of primary importance.

Real-time autonomous path planning is developed in many other fields than in the maritime area: underwater vehicles, aerospace, automotive, robotics, unmanned ground vehicles and virtual reality are some examples. Extending the literature review to other fields is helpful to have a better overview of the state-of-the-art.

Random sampling algorithms represent one of the state-of-the-art choices when dealing with real-time path planning of autonomous vehicles. Karaman and Frazzoli [14] stated that the introduction of random sampling algorithms had been the most significant step forward in motion planning of autonomous vehicles, and an exhaustive comparative analysis of various random sampling algorithms was provided. Among those, the Rapidly exploring Random Tree (RRT) algorithm is the most widely known: it was introduced by LaValle and Kuffner Jr. [15] for various applications, such as non-optimal motion planning of vehicles, manipulators, objects in very constrained spaces, spacecraft and virtual characters in video games. The optimising variant of RRT is named the RRT* [16]. The RRT and its variations are widely adopted for optimal path planning, including AUVs (Autonomous Underwater Vessels) [17–22], as they are relatively simple, fast and flexible. Zaccone and Martelli [23] presented in detail an RRT*-based approach to ship path planning for collision avoidance purposes, focusing on the offline, standalone behaviour of the algorithm, discussing the tuning of the optimisation parameters to obtain the best evasive manoeuvres in the presence of one or more obstacles. When dealing with autonomous navigation, online, real-time implementation of motion-planning systems is more complex than offline. Real-time applications of the PSO (Particle Swarm Optimisation) and RRT algorithms for marine vehicles were presented by Yan et al. [24] and Hernandez et al. [25], respectively, the last including experimental tests on a small AUV. Zaccone and Martelli [26] published preliminary online tests of an RRT*-based motion-planning algorithm applied to a large ship in a simulated environment. When dealing with large ships, COLREG compliance is a controversial and much discussed topic: COLREG, or Convention of International Regulations for Preventing Collisions at Sea [2], includes several rules and regulations for the proper conduction of marine vehicles. The most recent literature started to investigate how to incorporate COLREG into autonomous navigation algorithms. Zaccone et al. [27] proposed and tested offline a vector-based COLREG-compliant RRT* algorithm: such an approach was used and discussed in detail in this paper. Moreover, a different RRT-based approach to COLREG compliance was proposed by Chiang and Tapia [28], based on the concept of blocking areas. Other research results on COLREG-compliant autonomous navigation algorithms were published by Naeem et al. [29], Hu et al. [30], Lyu and Yin [31], Stankiewicz and Mullins [32].

This paper presents a Motion-Planning, Guidance and Control (MPGC) architecture for real-time autonomous navigation, with a particular focus on Optimal Path Planning (OPP) in the presence of fixed obstacles and other ships (also called targets in the rest of the paper). The presented MPGC architecture features an RRT*-based OPP module, capable of planning optimal COLREG-compliant and collision-free trajectories, efficiently managing multiple moving targets and fixed obstacles. The MPGC architecture integrates

the proposed OPP system with a Guidance and Control (GC) system [33,34] to control the motion of a high-fidelity ship 3DOF (Three Degrees Of Freedom) simulation model [35,36] in a challenging simulated environment. The simulation environment's main features were real-time simulation, fixed obstacles, dynamic moving targets and noisy sensors. Firstly, the paper introduces, breaks down and discusses the MPGC; secondly, it gives details on the RRT* algorithm and the optimal path problem formulation, focusing on path optimality, the definition of the cost function and constraints, obstacle management and COLREG compliance. Thirdly, the paper details the implementation of the proposed architecture and the integration with the GC system and simulation environment. The developed MPGC system was eventually tested in three realistic scenarios, and the results are critically discussed.

2. Motion-Planning, Guidance and Control System Architecture

This section aims to illustrate the layered architecture of the proposed autonomous MPGC system. The following sections focus on the paper's principal object, the OPP module. The main task of the MPGC navigation system is to allow a vessel to move in space with a certain degree of autonomy, i.e., without human intervention, at least in some operating conditions, from the perspective of a complete artificial intelligence-based decision-making system, capable of navigating the vessel during the whole mission.

The architecture presented in this paper mainly addressed the open-water navigation condition, characterised by some significant features. First, the own ship is supposed to move at a medium to high speed, i.e., the speed vector's drift component is significantly smaller than the longitudinal component. Moreover, the own ship is expected to maintain a reasonable distance from the other vessels in the scenario, a minimum of some ship lengths or twice the tactical diameter. Eventually, fixed obstacles and other crafts can be found in the scenario. Still, there is room for manoeuvring without reducing the ship's speed, which is supposed to be maintained constant. In everyday ship navigation, steering is usually considered the first choice solution for collision avoidance: ships, compared to wheeled vehicles, are better at steering than at braking, so the speed is reduced if the steering action alone is not sufficient.

Figure 1 represents the architecture of the MPGC system. During navigation, the situational awareness module collects information about the environment via sensors; the information related to the obstacles, in particular, is organised into a data structure by the obstacle management module and made available to the OPP module. The OPP module is the core of this paper: based on the path optimisation parameters, the position and speed feed-backs from the ship and the information on the fixed and moving obstacles detected in the environment, it computes the optimal, checking collision-free, COLREG-compliant path to reach the arrival point. The path is then processed by the GC system, which computes the appropriate control actions. Eventually, the control actions allow the actuators governing the ship to follow the planned path.

The MPGC system acts at discrete time intervals: with a preset re-planning rate, the obstacle data and the own ship's position and speed are updated, and the optimal path is recomputed and sent to the GC module. The OPP module predicts the motion of the detected moving obstacles to ensure collision avoidance and COLREG compliance. The periodic re-planning allows updating the obstacle's data with any unpredictable changes.

The situational awareness module was not the aim of this discussion. Thus, information about the surrounding ships and obstacles was assumed to be available from nautical charts, AISs and on-board sensors such as the radar. The paper discusses the obstacle management and OPP modules, focusing on the path optimality and feasibility and collision avoidance and COLREG compliance constraints. An RRT* algorithm efficiently computes the optimal path relying on fast nearest neighbour search algorithms to manage the obstacles. Eventually, the GC module was not the paper's primary target; however, it had a crucial role in the presented architecture. Later in the article, the described MPGC

system is tested in a simulation framework, including the described OPP module, a GC system and a three-degree-of-freedom ship simulation model.

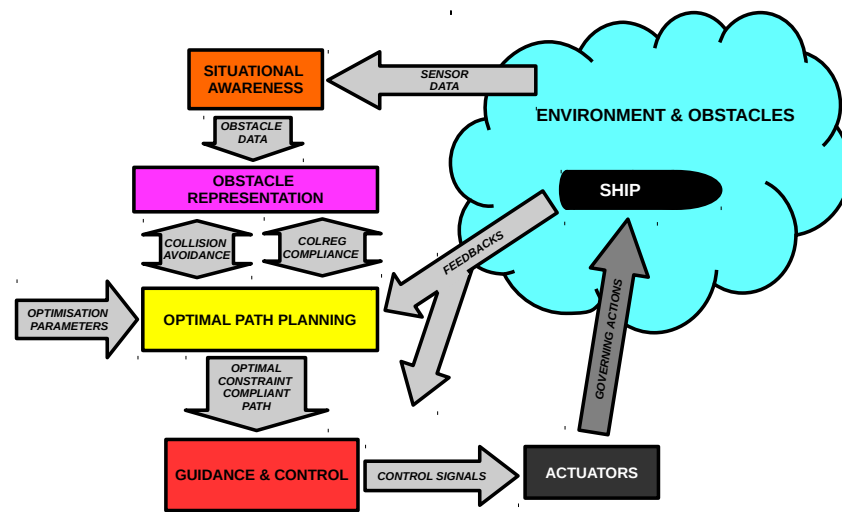


Figure 1. The guidance navigation and control architecture object of this paper.

3. The RRT* Path-Planning Algorithm

This section aims to give an overview of the RRT* algorithm [14–16], here used to compute the optimal path in the OPP module. RRT* is an optimised variation of the regular RRT algorithm: while the purpose of the RRT algorithm is to quickly generate feasible, collision-free trajectories, efficiently managing obstacles and forbidden regions, RRT* tries to minimise a cost function in the path-generation process, to provide heuristically optimal trajectories. The RRT and RRT* can be described as random-sampling algorithms, as they randomly sample the spatial domain, growing a tree that links the feasible configurations. While the mentioned references provided a detailed description of the algorithms, Figure 2 shows a graphical example of the RRT* step, which is described here. A tree G , composed of a set of nodes V and a set of edges E , is grown. At each step, a new node and edge are added according to the following actions:

- A random position x_{rand} is generated, then the node $x_{near} \in G$ that is nearest to x_{rand} is selected;
- A new node x_{new} is generated by moving towards x_{rand} from x_{near} by a fixed step;
- The edge linking x_{near} and x_{new} is then checked for collisions; both the node and the edge are then added to the tree.

The standard RRT algorithm is high in speed due to the limited computations; yet, it may generate sub-optimal, very irregular paths. Moreover, increasing the number of iterations does not improve the smoothness of the obtained trajectories. Note that constraints can be checked while generating the new edge, including, for example, a kinematic or dynamic model of the vehicle. The RRT* is an improved, optimality-driven version of the RRT that allows computing heuristically optimal paths according to a cost function. The cost function allows the algorithm to compare alternative path solutions, driving the tree growth towards optimality. After having generated the node x_{new} , the RRT* adds two additional steps to achieve path optimisation:

- A neighbourhood σ of the new node x_{new} is searched for the best parent node that minimises the cost function in x_{new} , in compliance with the constraints.
- x_{new} is considered a possible parent node of each of the neighbour nodes: in case x_{new} is a better parent than the current node, it becomes the new parent of the neighbour node. This action is usually called rewiring.

As mentioned above, the cost function allows controlling the tree's shape, i.e., the algorithm's output trajectories. Moreover, the solution heuristically converges to optimality at each iteration; thus, more iterations lead to better solutions.

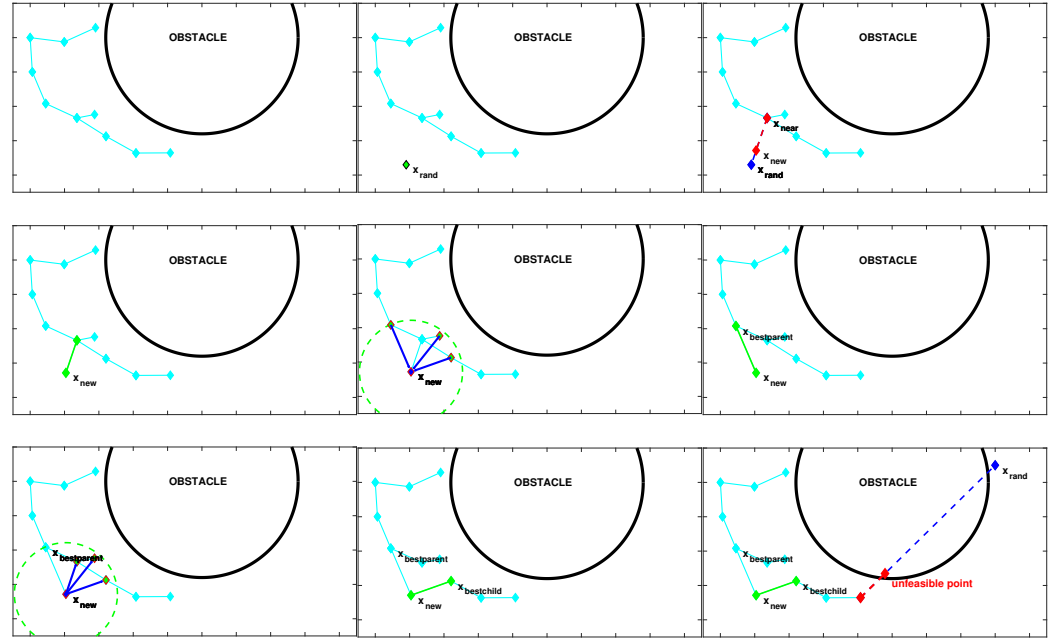


Figure 2. An example of how the RRT* algorithm works.

4. Optimal Path-Planning Problem Setup

This section discusses in detail the features of the optimal path-planning system. In particular, it defines the path structure; then, it discusses the cost function and constraints of the optimality problem, with a specific focus on the COLREG compliance constraint.

4.1. Path Structure and Turning Angle Constraint

The OPP module aims to plan a trajectory in the form of a set of waypoints, relying on the GC system for the actual path following. As opposed to dynamic model-based motion planning, this approach is a simple way to deal with the ship's complex, unpredictable dynamic behaviour. It requires only basic kinematic and geometric constraints to model the vessel's turning capabilities, relying on the control system to manage the model simplifications' effects. To this end, the planned path is represented by a set S of waypoints \mathbf{x}_p :

$$S = \{\mathbf{x}_p\}, p = 0, \dots, N \quad (1)$$

Two waypoints identify one route segment. The steering angle between two segments is related to the ship's manoeuvring capabilities. Expressing the p th and the $p + 1$ th segments in the vectorial forms $\mathbf{x}_p - \mathbf{x}_{p-1}$ and $\mathbf{x}_{p+1} - \mathbf{x}_p$, respectively, the steering angle in \mathbf{x}_p is:

$$|\theta_p| = \arccos \left(\frac{(\mathbf{x}_p - \mathbf{x}_{p-1}) \cdot (\mathbf{x}_{p+1} - \mathbf{x}_p)}{|\mathbf{x}_p - \mathbf{x}_{p-1}| |\mathbf{x}_{p+1} - \mathbf{x}_p|} \right) \quad (2)$$

In the proposed OPP system, such an angle is required to be less than a threshold value to be compatible with the vessel's manoeuvring capabilities.

4.2. Obstacle Management and Collision Avoidance

Two types of obstacles can be identified: fixed and moving. Fixed obstacles include anything that is not moving in the scenario, such as the shoreline or any floating yet not moving body. A set F of points (vectors) models the fixed obstacles. As the algorithm

will repeatedly compute the distance to the closest obstacle, organising the points in an appropriate data structure to make this operation faster can significantly enhance the computation performance. In this work, a kd-tree data structure [37] was implemented to ensure fast nearest-neighbour search capabilities. Moving obstacles, also referred to as targets, include any vessel moving in the scenario. The targets' instant positions are denoted $\mathbf{y}_i(t)$, while the set of all the target positions detected at time t is indicated by $M(t)$. Targets can in principle move according to any law of motion: for the purposes of this work, the targets' positions $\mathbf{y}_i(t_{rep})$ and speed vectors $\mathbf{w}_i(t_{rep})$ are collected and updated each time t_{rep} the OPP algorithm is called, at a defined path re-planning rate. Within the re-planning instants, the position of each target is extrapolated by approximating its motion as follows:

$$\mathbf{y}_i(t) = \mathbf{y}_i(t_{rep}) + \mathbf{w}_i(t_{rep})(t - t_{rep}) \quad (3)$$

When computing the evasive manoeuvre, the distance from the closest moving or fixed obstacle needs to be greater than a threshold value d_{safety} representing the distance below which the collision hazard is too high:

$$\min_{\mathbf{y} \in (F \cup M(t))} |\mathbf{y} - \mathbf{x}(t)| \geq d_{safety} \quad (4)$$

The safety distance should be defined considering the uncertainties of the obstacle detection system and the possible track errors due to the GC system.

4.3. Cost Function

The definition of the cost function is crucial to control the path shape and features: while the RRT grows, the optimising action of the RRT* algorithm locally reshapes the tree to minimise the desired cost. Three main desirable features have been identified:

- a limited steering action
- a reasonable margin from the (fixed or moving) obstacles
- a short path elongation

In principle, these features can be in conflict, and one or more of them might not be required: the desired result can be obtained as a trade trade-off solution via cost function aggregation, i.e., using a properly tuned weighted summation of different contributions. The general cost formulation at the generic waypoint \mathbf{x}_p can be expressed as follows:

$$c(\mathbf{x}_p) = \gamma_s c_s(\mathbf{x}_p) + \gamma_f c_f(\mathbf{x}_p) + \gamma_m c_m(\mathbf{x}_p) + \gamma_l c_l(\mathbf{x}_p) \quad (5)$$

where γ . represent the summation coefficients, c . are the cost contributions and the subscripts s, f, m, l indicate the steering action, the margin from fixed and moving obstacles and the path elongation, respectively. The cost c_s associated with the steering action is defined as follows:

$$c_s(\mathbf{x}_p) = \sum_{k=1}^{p-1} \theta_k^2 \quad (6)$$

The minimisation of such a cost function provides relatively smooth trajectories, with reasonably low path elongation. The minimisation of the control actions' squared sum is a common practice in many optimal control applications. Using solely this contribution produces reasonable solutions, so it is the suggested approach if there are no other particular trajectory requirements. The cost contributions associated with the obstacles, c_f and c_m , aim to model a repulsive effect of the obstacles in analogy with potential field methods, in order to selectively maximise the distance:

$$c_f(\mathbf{x}_p) = \sum_{k=0}^p \left(\min_{\mathbf{y}_i \in F} |\mathbf{y}_i - \mathbf{x}_k| \right)^{-1} \quad (7)$$

$$c_m(\mathbf{x}_p) = \sum_{k=0}^p \left(\min_{\mathbf{y}_i(t_k) \in M(t_k)} |\mathbf{y}_i(t_k) - \mathbf{x}_k| \right)^{-1} \quad (8)$$

Such cost functions allow generating very rounded trajectories, maximising the distance from the obstacles, at the cost of significant path elongation. Very constrained spaces, such as restricted waters, mitigate the elongation effect; otherwise, the use in combination with the path-shortening component is suggested. Eventually, the path-shortening component c_l allows minimising the path length, as it penalises the path elongation:

$$c_l(\mathbf{x}_p) = \sum_{k=1}^p |\mathbf{x}_k - \mathbf{x}_{k-1}| \quad (9)$$

This last component generates the shortest possible trajectories, linking straight lines with the narrowest turns allowed and barely respecting the safety distance. The summation weights selection should be made according to the path requirements to get the most appropriate solution for the specific ship, manoeuvring capabilities and mission requirement. Note eventually that all the cost contributions are non-negative: in other words, considering two waypoints \mathbf{x}_p and \mathbf{x}_q , if $p > q$ then $c(\mathbf{x}_p) \geq c(\mathbf{x}_q)$.

5. COLREG Compliance

The COLREG regulation aims to provide prescriptions and rules to avoid collisions between vessels. In particular, the regulation targets two main aspects: the vessel's lights and signals and the correct ship-master's behaviour when sailing in other ships' presence. In the steering and sailing rules, the COLREG regulation distinguishes three scenarios: crossing, overtaking and head-on. Referring to the ship navigation lights described in Figure 3, the required behaviour can be summarised as follows:

1. In the crossing condition, both ships can see one of the other's sidelights: in this case, the vessel that sees the other's red light must give way by turning to starboard, showing its red light (red-to-red).
2. In the overtaking condition, a vessel reaches another from the aft quarter: in this case, the overtaking vessels must not cross the overtaken vessel's route.
3. In the head-on condition, each vessel can see both the other vessel's sidelights. In this case, the rule prescribes both vessels to turn starboard, resulting in a red-to-red pass.

COLREG was written for human interpretation and based on typical human concepts, referring to ship light colours and relative concepts such as each ship's port or starboard side. Zacccone et al. [27] described a mathematical model of the COLREG Steering and Sailing rules for autonomous collision avoidance purposes. Such a model is designed to check if a given scenario is COLREG compliant or not, to be integrated into an OPP algorithm to generate COLREG-compliant trajectories. Note that while COLREG, most of the time, gives both vessels a prescription, the proposed algorithm suggests manoeuvring actions to the own ship only, relatively to the target vessel's position, heading and speed. Concerning Figure 4, the algorithm divides the collision scenario into three zones:

- The red texture represents a hazardous zone of radius d_{safety} , in which the probability of collision is too high: the purpose of the guidance system is to keep the obstacles out of this region.
- The yellow, delimited by a circle of radius d_{COLREG} , represents the area in which an unexpected action of the own ship or the obstacle vessel may lead to a hazardous situation. In this area, COLREG compliance is required.
- Outside the yellow zone, the vessels are considered sufficiently far apart (for example, far enough to be unable to see each other's lights), so the COLREG check is not required.

In other words, the MPGC system aims to keep the obstacles outside the red zone, respecting COLREG if they are in the yellow area.

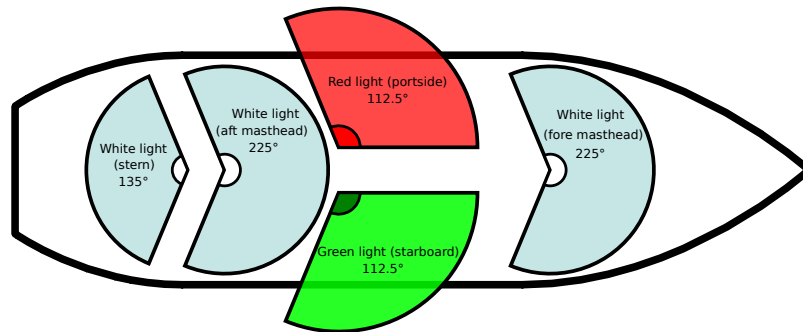


Figure 3. A ship's navigation lights as prescribed by COLREG.

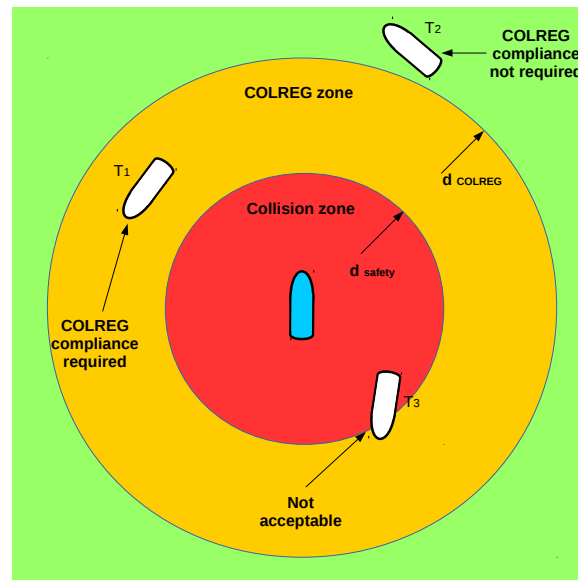


Figure 4. COLREG application zones: collision hazard zone (red), COLREG application zone (yellow) and free navigation zone (green).

5.1. Definitions

Referring to Figure 5, a right-hand NED (North-East-Down) system of coordinates $(\mathbf{e}_N, \mathbf{e}_E, \mathbf{e}_D)$ is defined. The own vessel's position is \mathbf{x}_0 , and its speed is \mathbf{v} ; the target vessel's position is \mathbf{y}_0 , and its speed is \mathbf{w} . As mentioned before, the considered scenario needs to match the following conditions:

- The distance between the two vessels must be greater than a threshold safety distance d_{safety} during all the manoeuvre.
- The initial distance between the vessels must be less than a convenient threshold $d_{COLREG} > d_{safety}$, otherwise no COLREG check is performed.

The above-mentioned conditions can be written as follows:

$$\begin{cases} d(\mathbf{y}(t), \mathbf{x}(t)) > d_{safety} \\ d(\mathbf{y}_0, \mathbf{x}_0) < d_{COLREG} \end{cases} \quad (10)$$

where $d(\cdot, \cdot)$ represents the distance between two points and $\mathbf{y}(t)$ and $\mathbf{x}(t)$ are the target ship's and the own ship's instant positions, respectively.

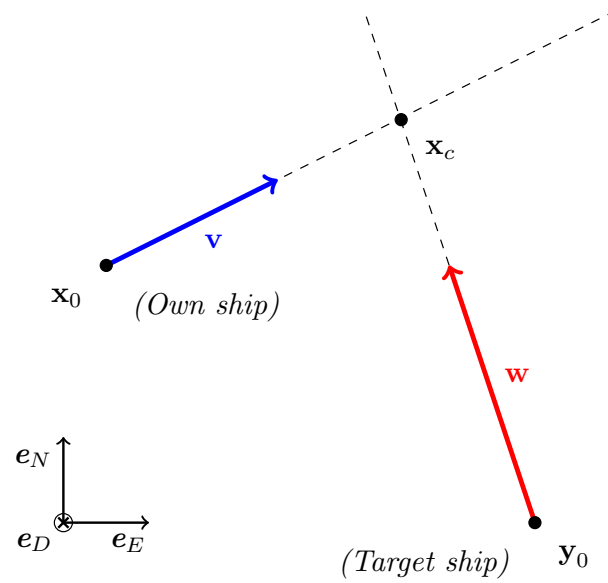


Figure 5. Symbols used to describe the considered scenarios.

5.2. Intersection Condition

If \mathbf{u} and \mathbf{r} are two vectors, perpendicular to \mathbf{v} and \mathbf{w} , respectively, a crossing condition can happen if the following linear system has exactly one solution:

$$\begin{bmatrix} \mathbf{u} \\ \mathbf{r} \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{u} \cdot \mathbf{x}_0 \\ \mathbf{r} \cdot \mathbf{y}_0 \end{bmatrix} \quad (11)$$

where $\mathbf{x} = (x, y)$ indicates a generic point. The condition can thus be written as follows:

$$\det \begin{pmatrix} \mathbf{u} \\ \mathbf{r} \end{pmatrix} \neq 0 \quad (12)$$

In order to avoid the strong equality for numeric purposes, the following equivalent crossing condition is preferred:

$$1 - \frac{|\mathbf{v} \cdot \mathbf{w}|}{|\mathbf{v}||\mathbf{w}|} > \varepsilon \quad (13)$$

where ε is a convenient tolerance threshold. The solution of the linear system (11) is the intersection point \mathbf{x}_c between the own ship's and the target's directions (Figure 5). There is a crossing scenario if \mathbf{x}_c is located both in front of the own ship and the target, and the following inequalities are verified:

$$\begin{cases} (\mathbf{x}_c - \mathbf{x}_0) \cdot \mathbf{v} > 0 \\ (\mathbf{x}_c - \mathbf{y}_0) \cdot \mathbf{w} > 0 \end{cases} \quad (14)$$

Moreover, it is possible to determine which vessel is engaging the point \mathbf{x}_c first, comparing the times of arrival:

$$\frac{|\mathbf{x}_c - \mathbf{x}_0|}{|\mathbf{v}|} - \frac{|\mathbf{x}_c - \mathbf{y}_0|}{|\mathbf{w}|} \begin{cases} > 0 \rightarrow \text{target ship first at } \mathbf{x}_c \\ = 0 \rightarrow \text{collision} \\ < 0 \rightarrow \text{own ship first at } \mathbf{x}_c \end{cases} \quad (15)$$

Note that in this case, equality means that there is a perfect collision.

5.3. Overtaking

According to COLREG, there is an overtaking scenario if the own ship is engaging the target ship from an angle θ_o of less than 22.5° from the own ship's aft:

$$\frac{\mathbf{v} \cdot \mathbf{w}}{|\mathbf{v}||\mathbf{w}|} > \cos(\theta_o) \quad (16)$$

If the inequality (16) is verified, COLREG requires the overtaking vessel not to cross the overtaken one's route. More in detail, if the two vessels' routes cross (Inequalities (13) and (14) are verified), the scenario is compliant if the target vessel engages the interception point first, in accordance with Equation (15).

5.4. Crossing

A crossing scenario satisfies Equations (13) and (14), while not satisfying (16): according to COLREG, in this case, the relative orientation between the ships approaching the intersection point needs to be described, by discussing the cross-product between \mathbf{v} and \mathbf{w} , (positive downwards):

$$\mathbf{v} \wedge \mathbf{w} \cdot \mathbf{e}_D \begin{cases} > 0 \rightarrow \text{own ship can see target's green} \\ < 0 \rightarrow \text{own ship can see target's red} \end{cases} \quad (17)$$

where the equality is excluded by Equation (13).

The combination of Equations (17) and (15) gives the following COLREG-compliant condition for a crossing scenario:

$$(\mathbf{v} \wedge \mathbf{w} \cdot \mathbf{e}_D) \left(\frac{|\mathbf{x}_c - \mathbf{x}_0|}{|\mathbf{v}|} - \frac{|\mathbf{x}_c - \mathbf{y}_0|}{|\mathbf{w}|} \right) < 0 \quad (18)$$

5.5. Head-On

In the head-on condition, COLREG prescribes a turn to starboard for both vessels. As the described algorithm plans the own vessel's manoeuvres only, if the angle between the vessel's routes is greater than a head-on threshold angle θ_h , the target vessel must be found on the own ship's port-side:

$$\mathbf{v} \wedge (\mathbf{y}_0 - \mathbf{x}_0) \cdot \mathbf{e}_D < 0 \quad (19)$$

If Inequality (19) is verified, the algorithm must check the crossing condition as well. Unlike in the case of the overtaking angle θ_o , COLREG does not give an exact value of the head-on angle θ_h . For this work, a value of 22.5° from the own ship's fore is assumed as reasonable.

6. Simulation Virtual Test Bed

The OPP algorithm described in the previous sections was implemented: as mentioned above, the RRT* algorithm computes the optimal path, and a kd-tree structure manages obstacles efficiently. The C programming language and gcc compiler were used to maximise the computation speed. It is worth mentioning that all the source code was custom developed and optimised, using only C standard libraries, to keep complete control of the planning process.

The OPP module was tested in a simulation setup, customised to match the testing requirements. The ship 3DOF simulation model was implemented in a MATLAB-Simulink environment coupled with an advanced GC system that allowed high-performance trajectory tracking and speed control. The OPP module was decoupled from the simulation to make the simulation more realistic. In other words, while the OPP algorithm is computing a new trajectory, the controlled ship is moving and following the old path. This feature was achieved by adopting the client-server architecture shown in Figure 6: the OPP algorithm was integrated into a standalone server, which plans the trajectory on request, after having

received the required information, and sends it back to the client. The implemented OPP module was either set to include or neglect the COLREG compliance constraint. When COLREG compliance is required, the algorithm tries to find a compliant solution. If no path is found, the planning is repeated, relaxing the COLREG compliance constraint. On the simulator side, a path-planning client sends the path re-planning request at a fixed rate, waits for the response and updates the GC trajectory once it has been received. The request message contains the following information:

- Own ship's starting coordinates and heading;
- Coordinates and heading of the arrival point;
- Own ship's speed;
- Number of detected targets;
- Targets' detected position, speed and heading.

The response message contains the waypoints that allow the own ship to reach the required arrival point safely. The arrival point is reached within a range of the desired heading, allowing a possible integration of the proposed architecture in a higher level offline long-range route planning, with very far apart waypoints. For example, a weather routing system layer [12,13] could be implemented on top of the proposed MPGC system. For this research, UDP (User Datagram Protocol) allows the client-server communication. Note that, from the software point of view, the OPP module, the GC and simulation are entirely uncoupled. Moreover, the communicating modules can be run either on the same computer or on different machines on the same network. Moreover, the same architecture applies to a "real" ASV (Autonomous Surface Vessel) for experimental testing, as long as the connection is ensured.

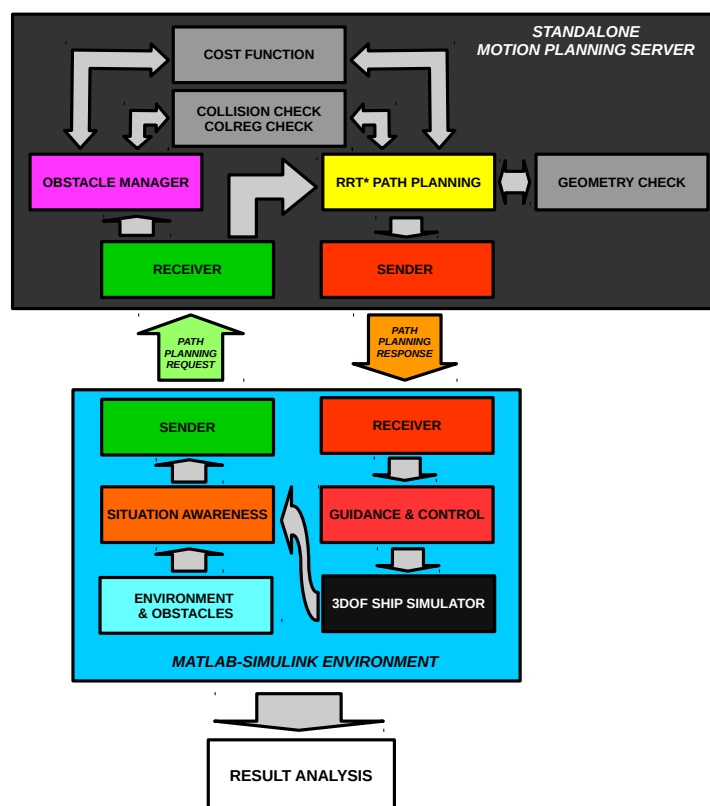


Figure 6. The implemented testing architecture.

The simulation model used in this work was a multi-domain platform, modelling the dynamics of a vessel in three degrees of freedom, taking into account the complete propulsion system with automation effects. The simulator integrates differential and algebraic equations and response surfaces to represent ship propulsion, manoeuvrability and control systems. De-

tailed information about the simulation platform was discussed by Alessandri et al. [35], while the validation of the simulator using sea trial data was published by Donnarumma et al. [38]. The simulation platform included a virtual reality interface [36], with the main focus on developing decision support systems for collision avoidance purposes [23,26], as well as guidance and control algorithms. In particular, the GC module integrated into the simulator was described in detail by Alessandri et al. [33], Donnarumma et al. [34].

7. Results and Discussion

The described MPGC system was tested in a simulated scenario, including the own ship and two other target ships. The extension of the test case map was 100×100 ship lengths (L_{pp}). The ship's mission consisted of sailing through the map from one corner to the opposite while keeping a safe distance of 10 ship lengths from both targets and respecting the COLREG constraint. The starting point was located in $[0, 0]$, and the goal point was $[100L_{pp}, 100L_{pp}]$. The collision avoidance algorithm was set up to keep the distance between the ship and the obstacles larger than 10 ship lengths while minimising a convenient balance between the proposed cost functions: the coefficients were selected via trial and error. Three scenarios were set up to be challenging for the OPP algorithm.

1. The first scenario included two targets in open-water navigation: the first target came from the starboard side, the second from the port side;
2. The second scenario featured a first target coming from ahead and another one coming from the starboard side, in open-water navigation;
3. The third scenario featured a head-on condition with a target ship coming from ahead in a narrow canal.

Table 1 shows the target data in the considered scenarios in the NED coordinate system. The system was tested in the simulation in realistic conditions to evaluate the proposed approach's guidance capabilities. Firstly, the simulation was performed in real time, and noise was artificially added to the obstacle data to replicate actual sensors. A re-planning rate of 15 s was considered. On average, the time required for the re-planning was about 0.5 s. The simulation framework was set up on a standard commercial notebook.

It should be pointed out that, in the considered scenarios, the COLREG-compliant and non-compliant trajectories were significantly different. The result obtained considering and neglecting the COLREG constraint will be compared and discussed in the following section. Moreover, for each case, the targets' unexpected behaviour was simulated to stress the algorithm and test the MPGC system's reactive capabilities.

Table 1. Route and starting point of the targets in the simulated scenarios.

	Scenario 1	Scenario 2	Scenario 3
Type	Open water	Open water	Narrow canal
Fixed obstacles	No	No	Yes
Target 1 starting point	$[0L_{pp}, 100L_{pp}]$	$[90L_{pp}, 100L_{pp}]$	$[100L_{pp}, 100L_{pp}]$
Target 2 starting point	$[100L_{pp}, 25L_{pp}]$	$[40L_{pp}, 100L_{pp}]$	-
Target 1 route	315°	240°	225°
Target 2 route	135°	270°	-

7.1. Scenario 1

The first scenario featured two moving targets, respectively starting at $[0L_{pp}, 100L_{pp}]$ and $[100L_{pp}, 25L_{pp}]$ and sailing along routes 315° and 135° , respectively. The simulation results with and without the COLREG compliance requirement are shown in Figure 7 and Figure 8, respectively. Note how different the solutions were: if COLREG was required, the own ship was forced to give way to the first target, coming from the starboard side, while the COLREG non-compliant solution allowed maintaining more distance from the obstacles. Figure 9 presents a conveniently modified version of the scenario, featuring an unexpected change in the targets' course that forces the OPP algorithm to alter the intended

ship's path significantly. Note how the own ship suddenly steers to its starboard side to avoid the unexpected and inconvenient target's steering and is forced to steer again to port to avoid the second obstacle, changing from a COLREG-compliant stand-on behaviour to a non-compliant give-way. Figure 10a–c shows the DTC (Distance-To-Collision) from the targets during the simulated manoeuvres. Note that the simulated paths always respected the safety distance.

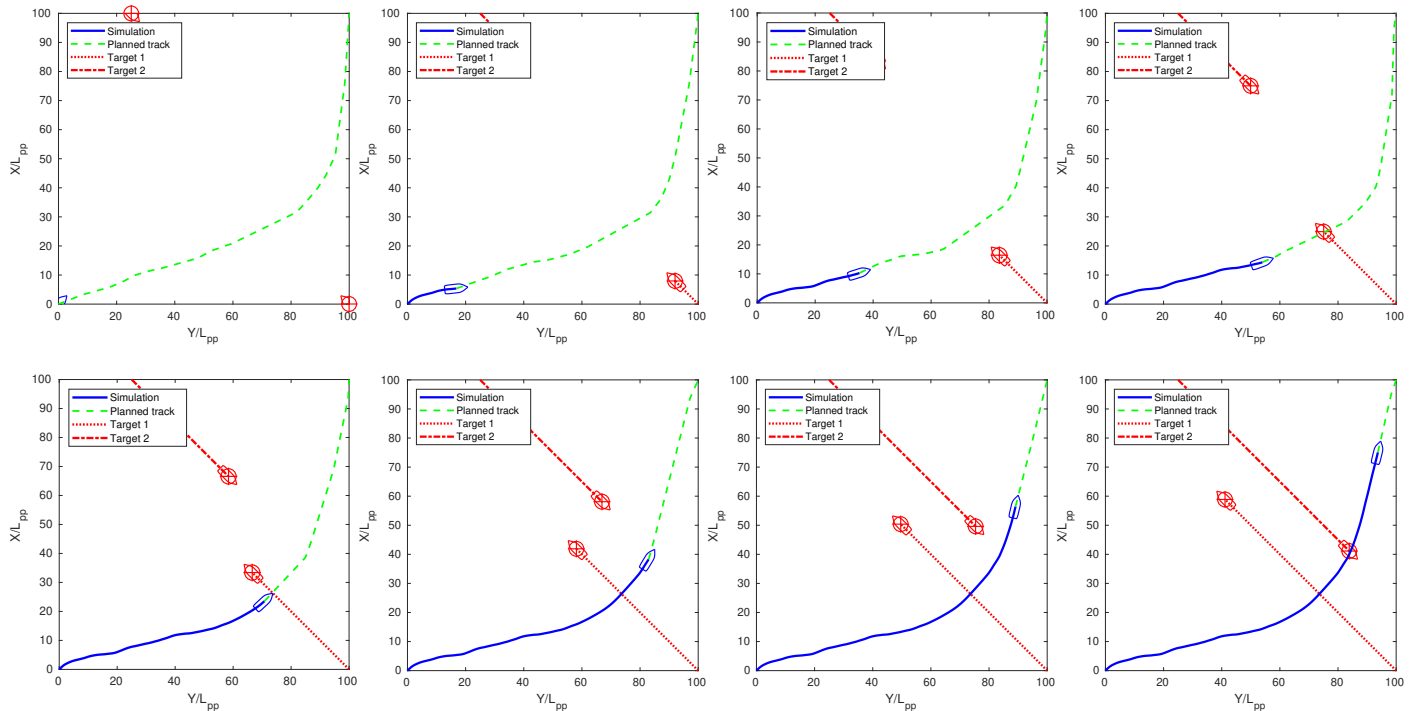


Figure 7. Scenario 1: simulation of a COLREG-compliant manoeuvre.

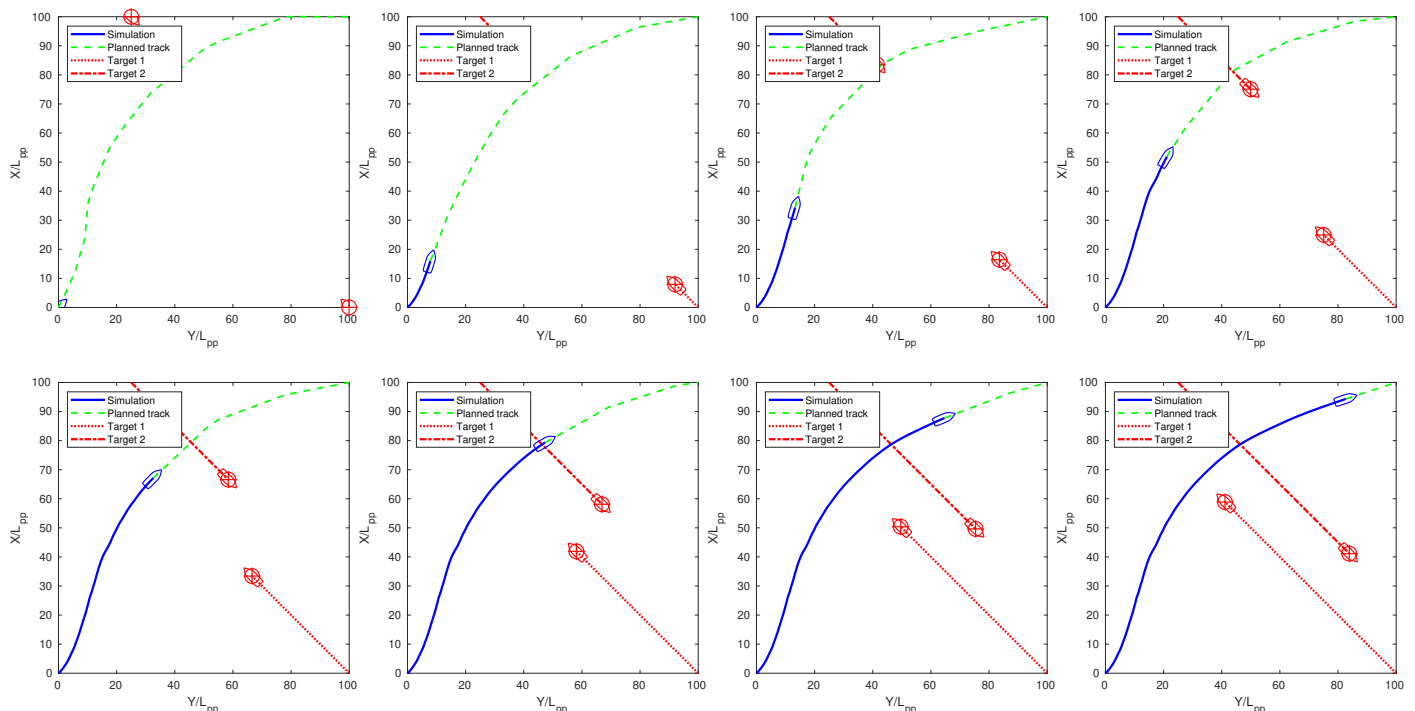


Figure 8. Scenario 1: simulation of a COLREG-non-compliant manoeuvre.

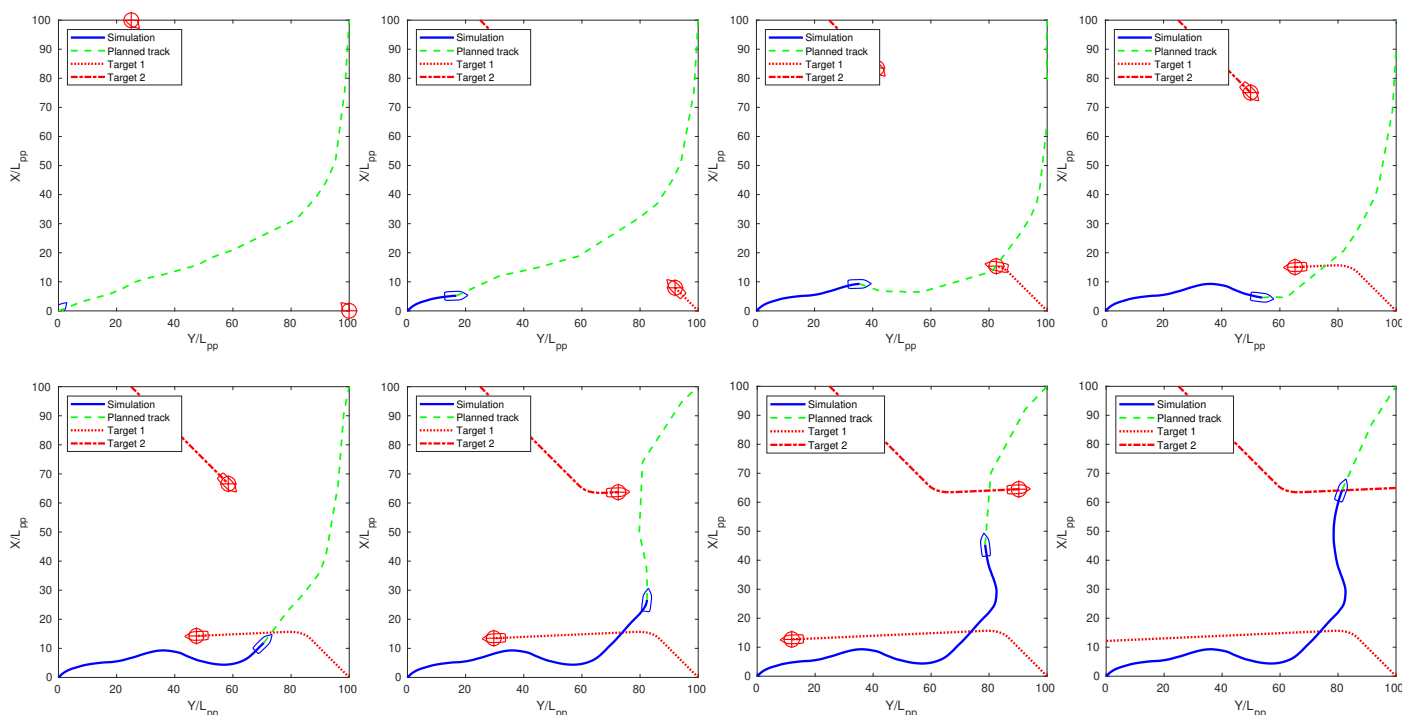


Figure 9. Scenario 1: simulation of a COLREG-compliant manoeuvre with unexpected target behaviour.

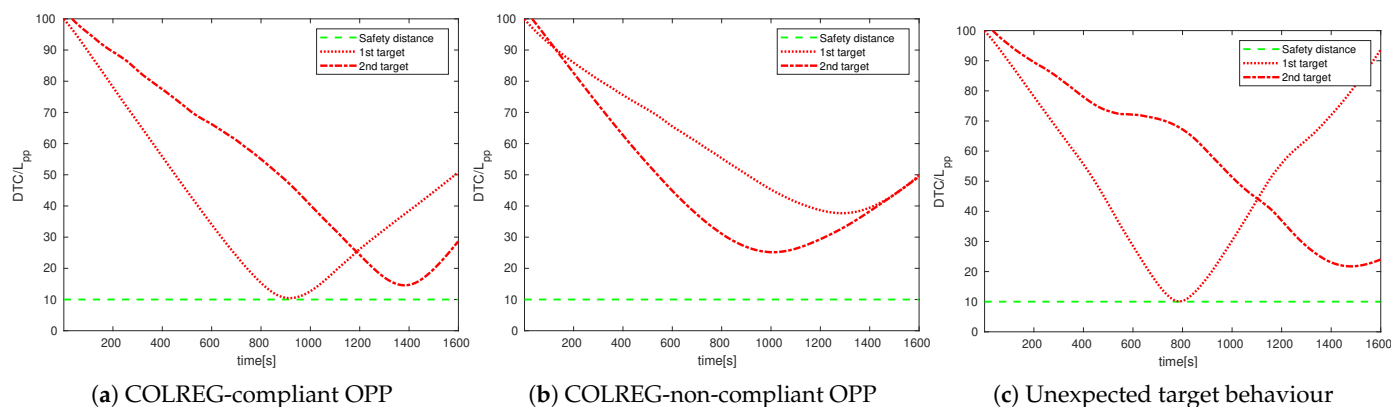


Figure 10. Scenario 1: DTC during the simulation of a COLREG-compliant manoeuvre (a), COLREG non-compliant manoeuvre (b) and COLREG-compliant manoeuvre with unexpected target behaviour (c).

7.2. Scenario 2

In the second scenario, the two moving targets started at $[90L_{pp}, 100L_{pp}]$ and $[40L_{pp}, 100L_{pp}]$, along routes 240° and 270° , respectively. The simulation results are again presented, including and excluding the COLREG compliance requirement. Figures 11 and 12 show the results: note that, in this second case, the non-compliant solution was quite close to the targets, as the OPP algorithm found a direct, yet hazardous solution, which was indeed not COLREG compliant because it did not give way to the obstacle coming from the starboard side. The scenarios presented in Figures 13 and 14 include, once again, an unexpected variation of the second target's course towards the own ship that forced the OPP algorithm to react to avoid a collision. In particular, a minor and a major course change were considered, respectively. Eventually, Figure 15 shows a further scenario variation where both the targets were close to the own ship and behaved unexpectedly. Repetitive path updates allowed the own ship to avoid the collision finally.

Figure 16 shows the DTC from the targets during the simulated manoeuvres. The safety distance was respected in both COLREG-compliant and non-compliant manoeuvres in the absence of targets' unexpected behaviour, as shown in Figure 16a,b. Moreover, the simulated trajectory respected the safety distance in the presence of a minor unexpected target's course change. In contrast, a major change made the own ship violate the safety distance by a small amount, as Figure 16c,d show. This slight violation was considered acceptable: the scenario was built with the explicit purpose of stressing the MPGC system's reactive path-planning features, and the target's steering was very unlikely in a real navigation scenario. Moreover, the safety distance should consider possible slight violations due to measurement errors and delays due to control and actuation systems. Eventually, the own ship was able to respect the safety distance in the last simulation, as shown in Figure 16e.

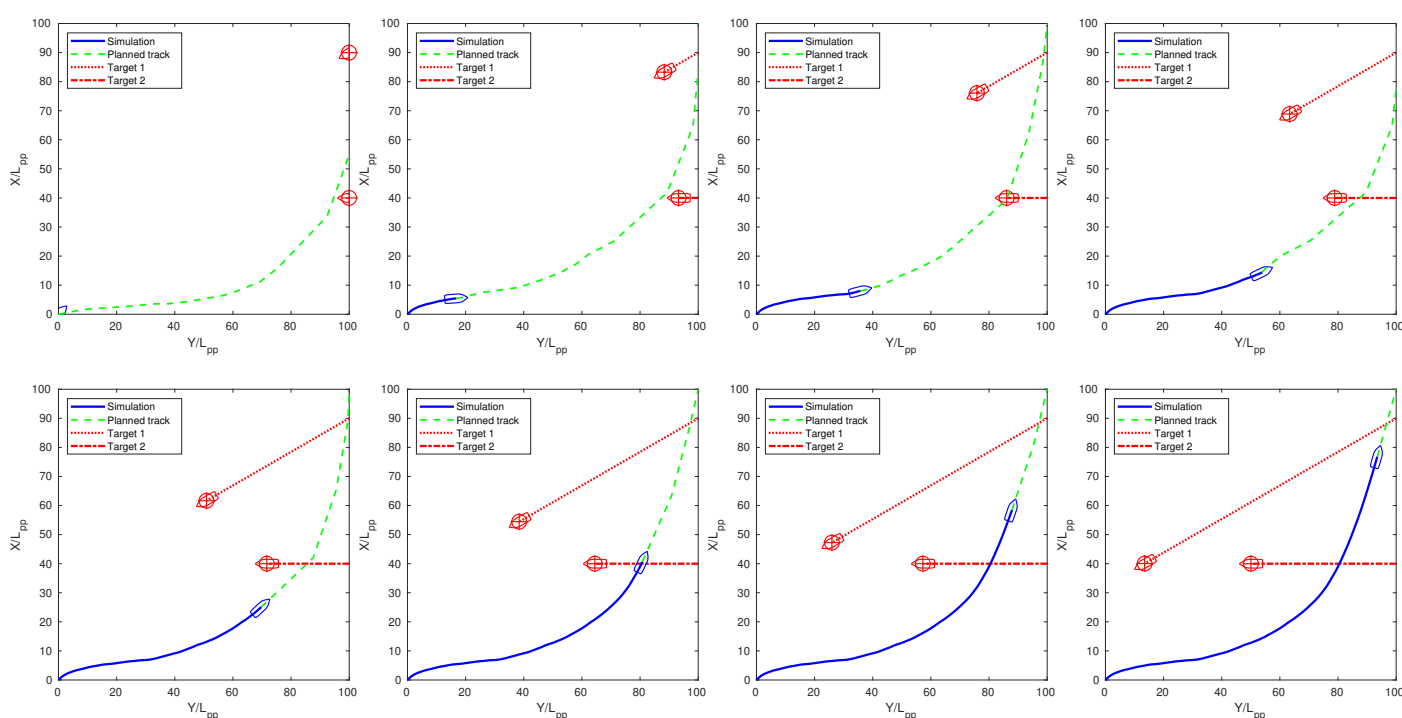


Figure 11. Scenario 2: simulation of a COLREG-compliant manoeuvre.

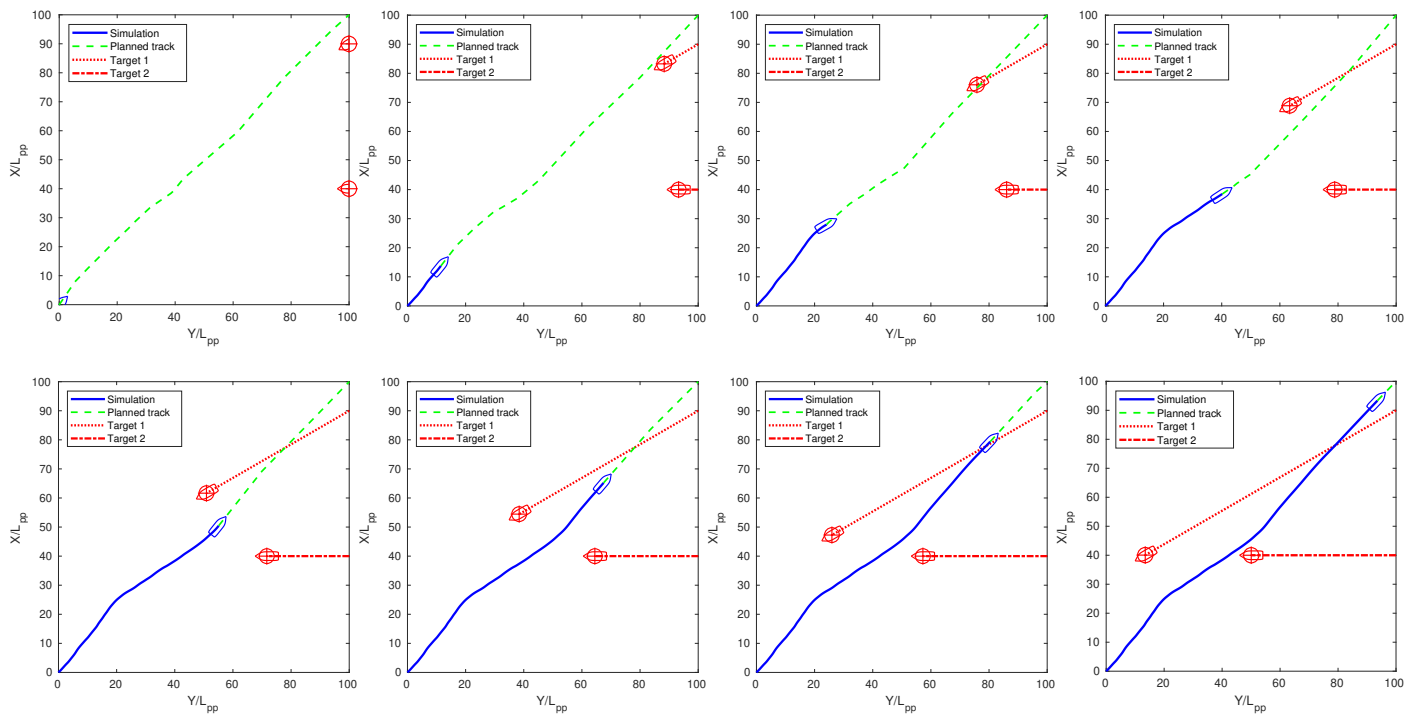


Figure 12. Scenario 2: simulation of a COLREG-non-compliant manoeuvre.

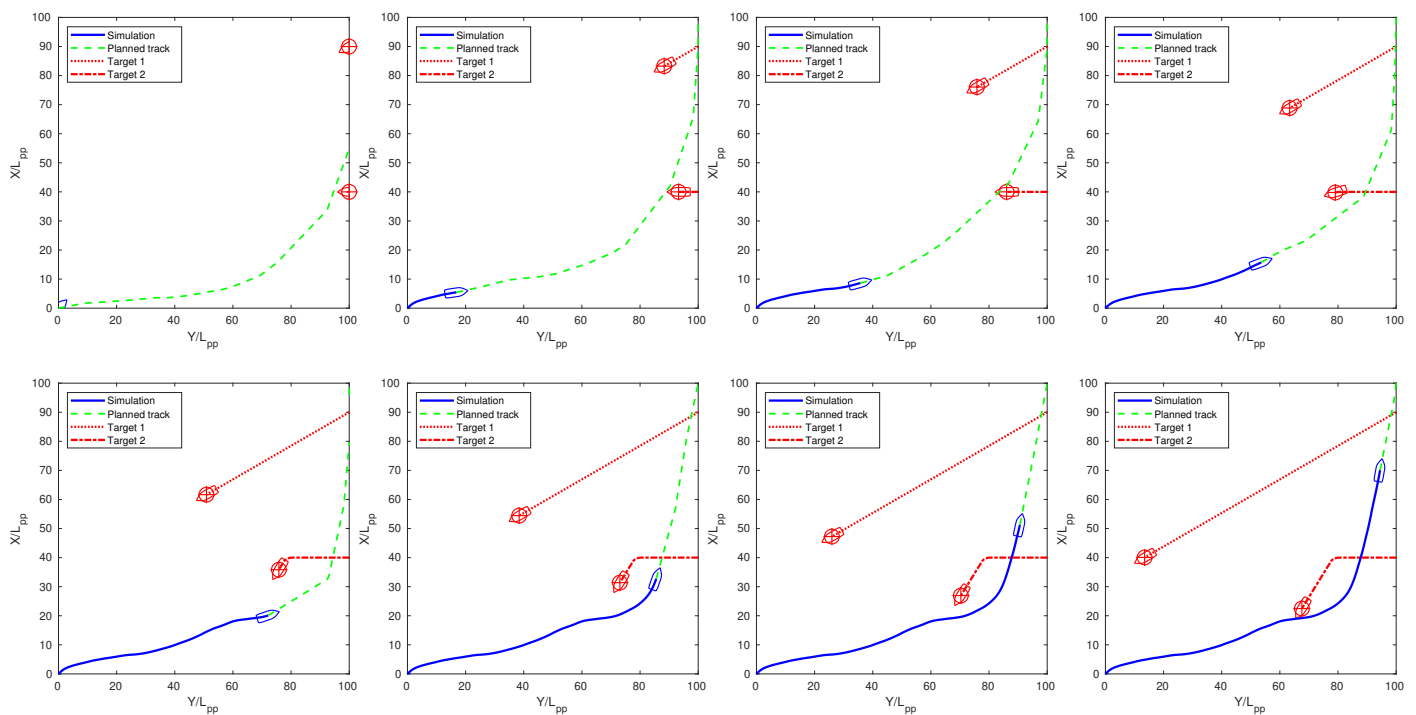


Figure 13. Scenario 2: simulation of a COLREG-compliant manoeuvre with unexpected target behaviour (minor course change).

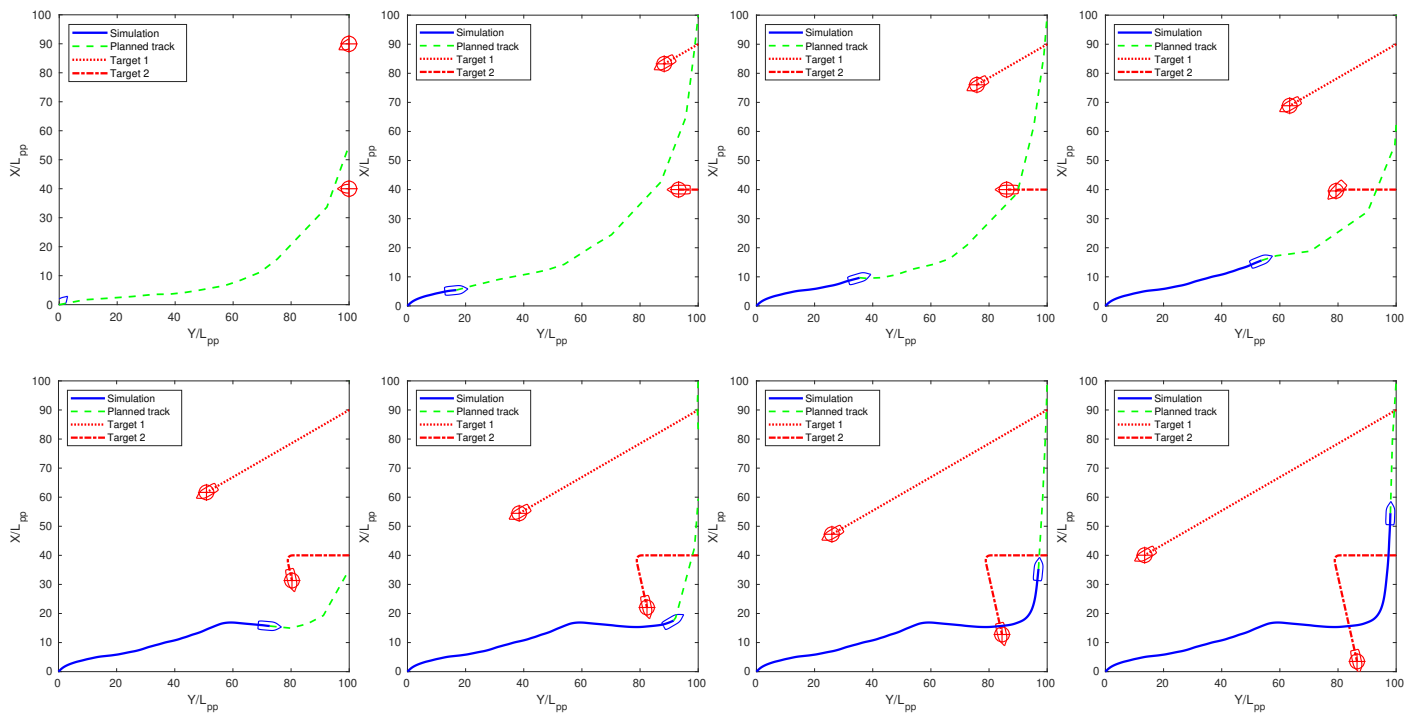


Figure 14. Scenario 2: simulation of a COLREG-compliant manoeuvre with unexpected target behaviour (major course change).

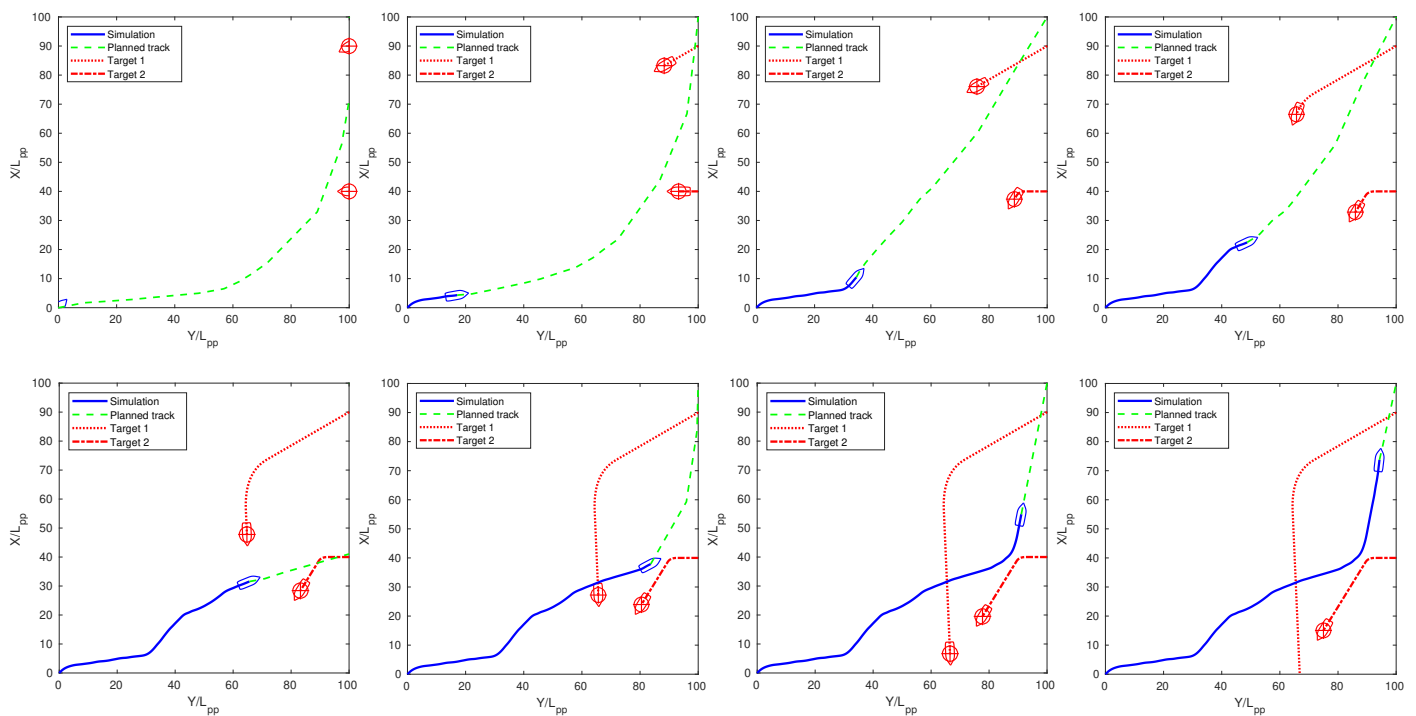


Figure 15. Scenario 2: simulation of a COLREG-compliant manoeuvre with unexpected target behaviour (both targets changing course).

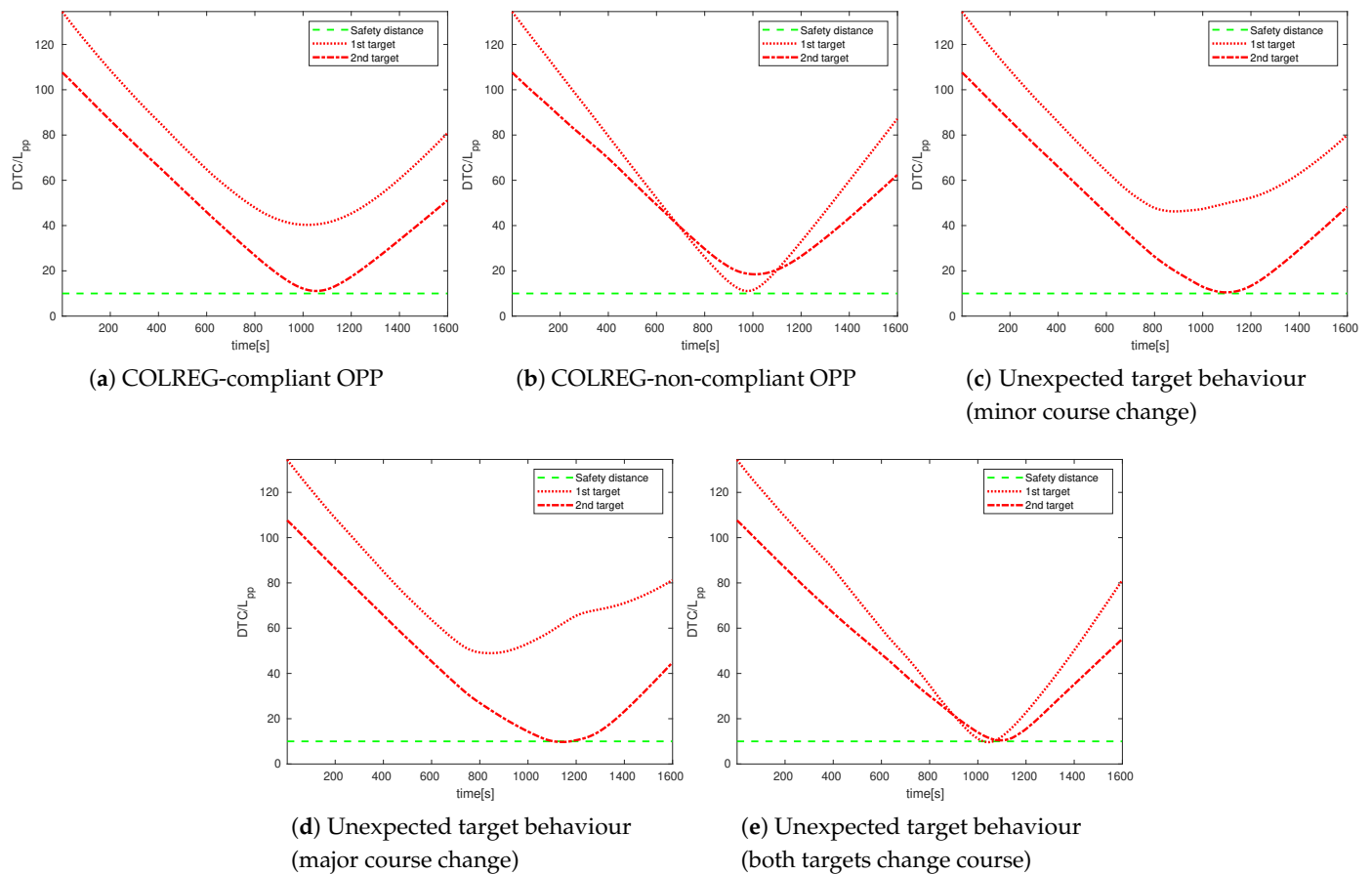


Figure 16. Scenario 2: DTC during simulation of a COLREG-compliant manoeuvre (a), COLREG-non-compliant manoeuvre (b) and COLREG-compliant manoeuvre with unexpected target behaviour (c–e).

7.3. Scenario 3

As opposed to the first two, the third scenario featured restricted waters, featuring both fixed obstacles (the canal walls) and one moving target. In particular, there was only one moving target, starting at $[100L_{pp}, 100L_{pp}]$ with a heading of 225° . Both the target and the own ship were sailing into a narrow canal in opposite directions. Note that the scenario was perfectly symmetrical; thus, neglecting COLREG compliance did not bring any significant advantage: the solutions might differ or not because the non-compliant RRT can randomly choose to steer to either side to avoid the collision. The simulation presented in Figure 17 was obtained using the COLREG-compliant algorithm, while the non-compliant path planning provided similar results. The third scenario was modified to stress the OPP module's re-planning capability by including an unexpected manoeuvre made by the target. The target's new route did not allow a COLREG-compliant path, and the OPP was forced to switch to non-compliant mode. The simulation results presented in Figure 18 highlight this behaviour. The DTC in Figure 19a,b also includes the distance from the fixed obstacles. Note that in Figure 16b, the path barely respected the safety distance because of the target's sudden route changes.

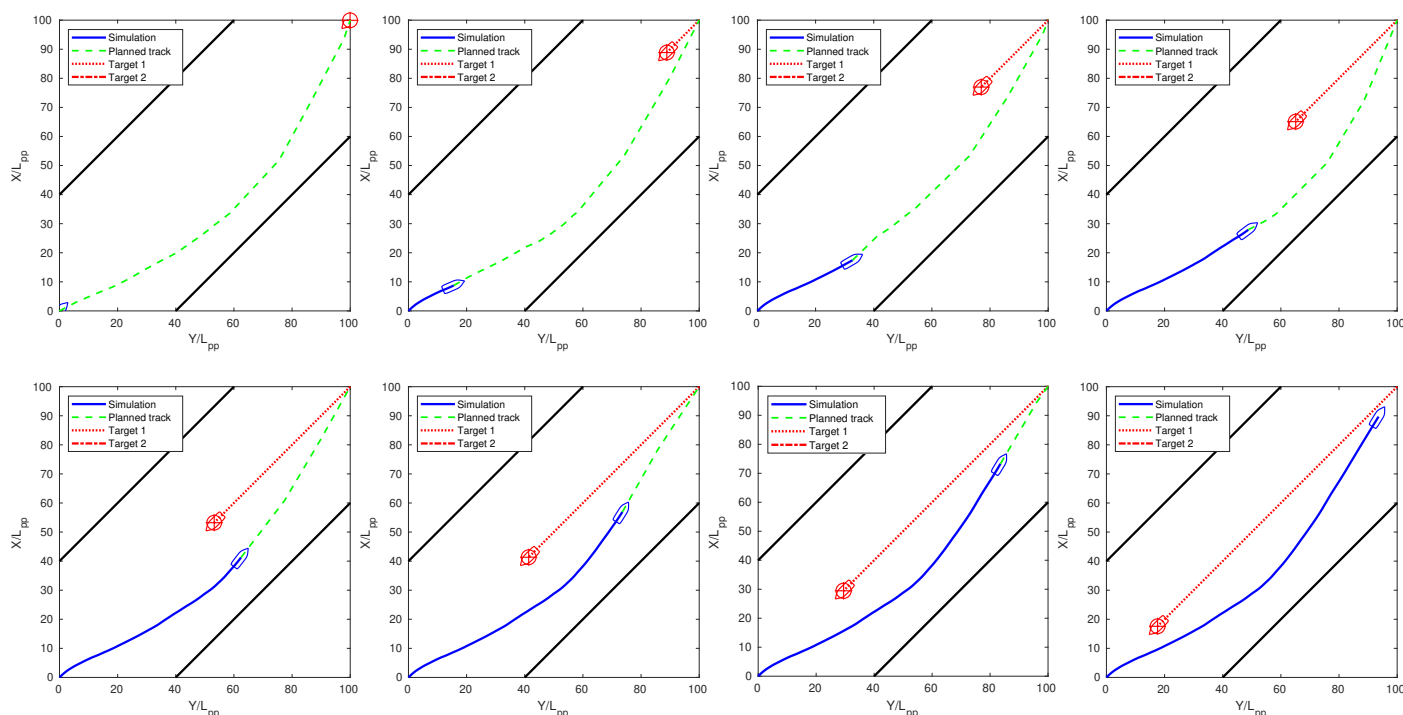


Figure 17. Scenario 3: simulation of a COLREG-compliant manoeuvre.

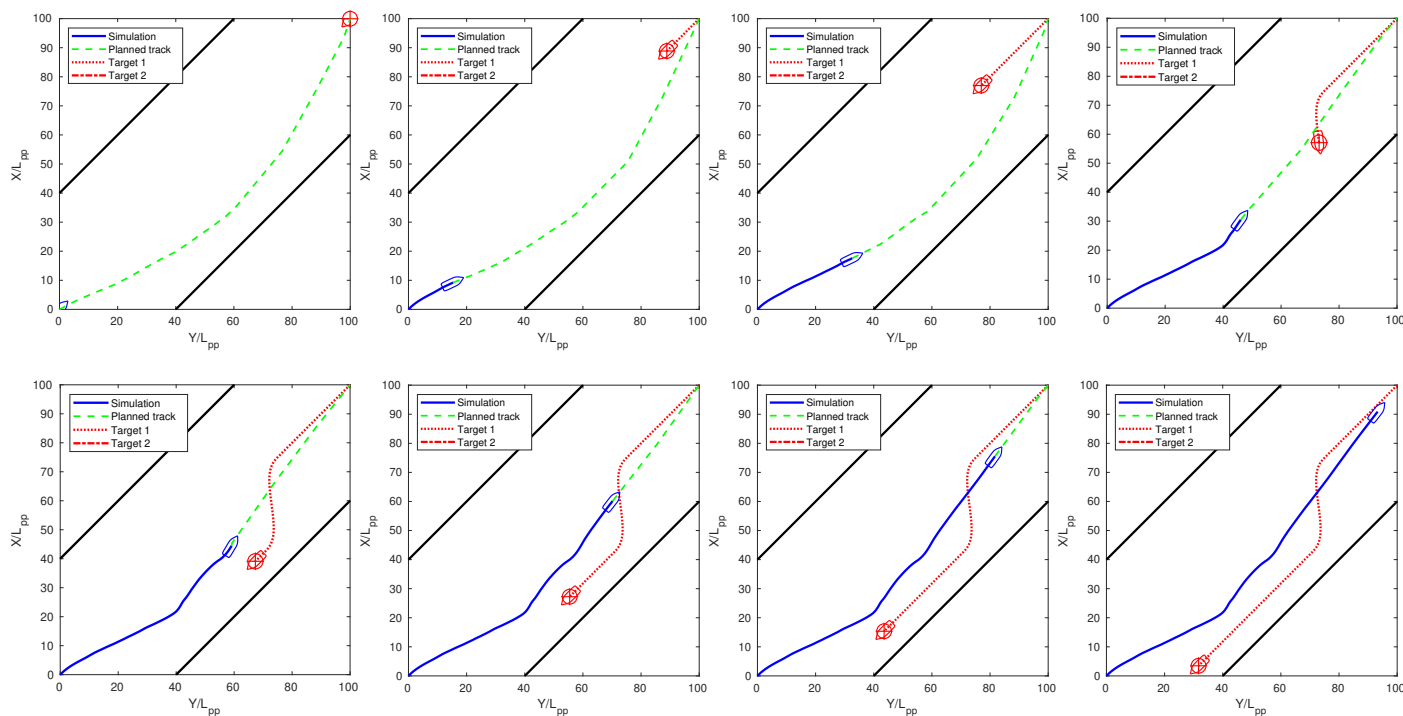


Figure 18. Scenario 3: simulation of a COLREG-compliant manoeuvre with unexpected target behaviour.

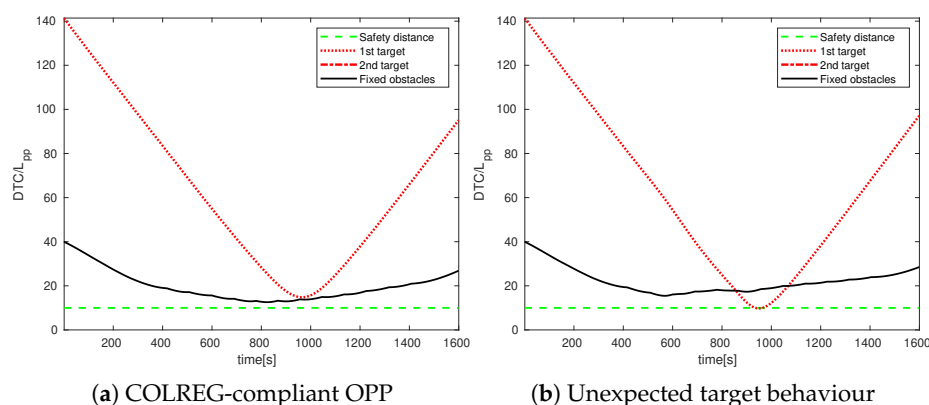


Figure 19. Scenario 3: DTC during simulation of a COLREG-compliant manoeuvre (Figure 16a) and COLREG-compliant manoeuvre with unexpected target behaviour (Figure 16c).

8. Conclusions

In this paper, an optimal path-planning system for ships' autonomous navigation, based on the RRT* algorithm, was discussed in detail, focusing on collision avoidance and COLREG compliance. A motion-planning, guidance and control architecture was proposed, integrating the OPP with a guidance and control module. The proposed MPGC architecture was implemented in a virtual simulation testbed to be tested in realistic scenarios. The MPGC system autonomously controlled a ship simulation model in real time in three realistic scenarios. The simulation featured both open and restricted navigation, multiple moving targets and fixed obstacles and simulated noisy sensors and moving targets changing their trajectory unexpectedly to put the OPP to the test. Simulations were performed both considering and neglecting the COLREG compliance constraint for comparison purposes.

The simulation results highlighted the described approach's potential: thanks to the limited time required by the OPP, the periodic re-planning showed good planning and reactive planning capabilities. The simulated trajectories respected the safety distances in most of the cases. In some of the simulations, unexpected targets' trajectory changes, deliberately designed to put a strain on the MPGC system, resulted in a safety distance violation by a small and acceptable amount. The MPGC system managed to keep control of the own ship and conclude the manoeuvre, switching from COLREG compliance to non-compliance when necessary. To summarise, the proposed architecture showed good potential for autonomous navigation applications.

However, many aspects need further investigation. On the simulation side, the MPGC system, which can be seen as a medium-range or short-range collision avoidance system, should be integrated with a higher level, offline route planning system, managing the long-range routing. Moreover, the weather's effects should be included in the simulation. Furthermore, the OPP system should be improved to provide optimal manoeuvres considering the current weather conditions. On the experimental side, a model-scale testbed will be implemented to perform tests on an autonomous ship model, including real hardware and actual sensor data.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The author declares no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

3DOF	Three Degrees Of Freedom
AIS	Automatic Identification System
ASV	Autonomous Surface Vessel
AUV	Autonomous Underwater Vessel
COLREG	Convention on the International Regulations for Preventing Collisions at Sea
DTC	Distance-To-Collision
EMSA	European Maritime Safety Agency
GC	Guidance and Control
MPGC	Motion-Planning Guidance and Control
NED	North-East-Down reference frame
OPP	Optimal Path Planning
PSO	Particle Swarm Optimisation
radar	radio detection and ranging
RRT	Rapidly exploring Random Tree
UDP	User Datagram Protocol

References

1. EMSA. *Annual Overview of Marine Casualties and Incidents*; European Maritime Safety Agency: Lisbon, Portugal, 2018.
2. International Maritime Organization. *COLREGs: Convention on the International Regulations for Preventing Collisions at Sea*; International Maritime Organization: London, UK, 1972.
3. Fang, M.C.; Tsai, K.Y.; Fang, C.C. A Simplified Simulation Model of Ship Navigation for Safety and Collision Avoidance in Heavy Traffic Areas. *J. Navig.* **2018**, *71*, 837–860. [\[CrossRef\]](#)
4. Montewka, J.; Goerlandt, F.; Kujala, P. Determination of collision criteria and causation factors appropriate to a model for estimating the probability of maritime accidents. *Ocean Eng.* **2012**, *40*, 50–61. [\[CrossRef\]](#)
5. Skjong, R.; Mjelde, K.M. optimal evasive manoeuvre for a ship in an environment of fixed installations and other ships. *Nor. Marit. Res.* **1984**, *12*, 15–22.
6. Ito, M.; Zhnng, F.; Yoshida, N. Collision avoidance control of ship with genetic algorithm. In Proceedings of the 1999 IEEE International Conference on Control Applications (Cat. No.99CH36328), Kohala Coast, HI, USA, 22–27 August 1999; Volume 2, pp. 1791–1796. [\[CrossRef\]](#)
7. Smierzchalski, R.; Michalewicz, Z. Modeling of ship trajectory in collision situations by an evolutionary algorithm. *IEEE Trans. Evol. Comput.* **2000**, *4*, 227–241. [\[CrossRef\]](#)
8. Alvarez, A.; Caiti, A.; Onken, R. Evolutionary Path Planning for Autonomous Underwater Vehicles in a Variable Ocean. *IEEE J. Ocean Eng.* **2004**, *29*, 418–429. [\[CrossRef\]](#)
9. Hasegawa, K.; Fukuto, J.; Miyake, R.; Yamazaki, M. An intelligent ship handling simulator with automatic collision avoidance function of target ships. In Proceedings of the INSLC 17—International Navigation Simulator Lecturers’ Conference Rostock-Warnemuende, Rostock, Germany, 3–7 September 2012; pp. 1–10.
10. Tam, C.; Bucknall, R.; Greig, A. Review of collision avoidance and path planning methods for ships in close range encounters. *J. Navig.* **2009**, *62*, 455–476. [\[CrossRef\]](#)
11. Panda, M.; Das, B.; Subudhi, B.; Pati, B.B. A Comprehensive review of path planning algorithms for autonomous underwater vehicles. *Int. J. Autom. Comput.* **2020**, *17*, 321–352. [\[CrossRef\]](#)
12. Zaccone, R.; Ottaviani, E.; Figari, M.; Altosole, M. Ship voyage optimization for safe and energy-efficient navigation: A dynamic programming approach. *Ocean Eng.* **2018**, *153*, 215–224. [\[CrossRef\]](#)
13. Vettor, R.; Guedes Soares, C. Development of a ship weather routing system. *Ocean Eng.* **2016**, *123*, 1–14. [\[CrossRef\]](#)
14. Karaman, S.; Frazzoli, E. Sampling-based algorithms for optimal motion planning. *Int. J. Robot. Res.* **2011**, *30*, 846–894. [\[CrossRef\]](#)
15. LaValle, S.; Kuffner, J., Jr. Randomized kinodynamic planning. *Int. J. Robot. Res.* **2001**, *20*, 378–400. [\[CrossRef\]](#)
16. Karaman, S.; Frazzoli, E. Incremental sampling-based algorithms for optimal motion planning. In *Robotics: Science and Systems*; MIT Press: Cambridge, MA, USA, 2011; Volume 6, pp. 267–274.
17. Li, Y.; Zhang, F.; Xu, D.; Dai, J. Liveness-based RRT algorithm for autonomous underwater vehicles motion planning. *J. Adv. Transp.* **2017**, *2017*, 7816263. [\[CrossRef\]](#)
18. Li, Y.; Cui, R.; Li, Z.; Xu, D. Neural network approximation based near-optimal motion planning with kinodynamic constraints using RRT. *IEEE Trans. Ind. Electron.* **2018**, *65*, 8718–8729. [\[CrossRef\]](#)
19. Li, J.; Yang, C. AUV Path Planning Based on Improved RRT and Bezier Curve Optimization. In Proceedings of the 2020 IEEE International Conference on Mechatronics and Automation (ICMA), Beijing, China, 13–16 October 2020; pp. 1359–1364.
20. Yu, L.; Wei, Z.; Wang, Z.; Hu, Y.; Wang, H. Path optimization of AUV based on smooth-RRT algorithm. In Proceedings of the 2017 IEEE International Conference on Mechatronics and Automation (ICMA), Takamatsu, Japan, 6–9 August 2017; pp. 1498–1502.

21. Heo, Y.J.; Chung, W.K. RRT-based path planning with kinematic constraints of AUV in underwater structured environment. In Proceedings of the 2013 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Jeju, Korea, 30 October–2 November 2013; pp. 523–525.
22. Wang, X.; Williams, S.; Angley, D.; Gilliam, C.; Jackson, T.; Ellem, R.; Bessell, A.; Moran, B. RRT* Trajectory Scheduling Using Angles-Only Measurements for AUV Recovery. In Proceedings of the 2019 22th International Conference on Information Fusion (FUSION), Ottawa, ON, Canada, 2–5 July 2019; pp. 1–6.
23. Zacccone, R.; Martelli, M. A random sampling based algorithm for ship path planning with obstacles. Revolutionary Technology Inspiring Ship Control. In Proceedings of the 14th International Naval Engineering Conference and Exhibition incorporating the 1st International Ship Control Systems Symposium, INEC/iSCSS 2018, Glasgow, UK, 2–4 October 2018; pp. 170–178. [\[CrossRef\]](#)
24. Yan, Z.; Li, J.; Wu, Y.; Zhang, G. A real-time path planning algorithm for AUV in unknown underwater environment based on combining PSO and waypoint guidance. *Sensors* **2019**, *19*, 20. [\[CrossRef\]](#) [\[PubMed\]](#)
25. Hernandez, J.D.; Vidal, E.; Vallicrosa, G.; Galceran, E.; Carreras, M. Online path planning for autonomous underwater vehicles in unknown environments. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 1152–1157.
26. Zacccone, R.; Martelli, M. A collision avoidance algorithm for ship guidance applications. *J. Mar. Eng. Technol.* **2020**, *19*, 62–75. [\[CrossRef\]](#)
27. Zacccone, R.; Martelli, M.; Figari, M. A COLREG-Compliant Ship Collision Avoidance Algorithm. In Proceedings of the IEEE European Control Conference—ECC2019, Naples, Italy, 25–28 June 2019; pp. 2530–2535.
28. Chiang, H.T.L.; Tapia, L. COLREG-RRT: An RRT-Based COLREGS-Compliant Motion Planner for Surface Vehicle Navigation. *IEEE Robot. Autom. Lett.* **2018**, *3*, 2024–2031. [\[CrossRef\]](#)
29. Naeem, W.; Irwin, G.W.; Yang, A. COLREGs-based collision avoidance strategies for unmanned surface vehicles. *Mechatronics* **2012**, *22*, 669–678. [\[CrossRef\]](#)
30. Hu, L.; Naeem, W.; Rajabally, E.; Watson, G.; Mills, T.; Bhuiyan, Z.; Salter, I. Colregs-compliant path planning for autonomous surface vehicles: A multiobjective optimization approach. *IFAC-PapersOnLine* **2017**, *50*, 13662–13667. [\[CrossRef\]](#)
31. Lyu, H.; Yin, Y. COLREGS-Constrained Real-Time Path Planning for Autonomous Ships Using Modified Artificial Potential Fields. *J. Navig.* **2019**, *72*, 588–608. [\[CrossRef\]](#)
32. Stankiewicz, P.G.; Mullins, G.E. Improving Evaluation Methodology for Autonomous Surface Vessel COLREGS Compliance. In Proceedings of the OCEANS 2019—Marseille, Marseille, France, 17–20 June 2019; pp. 1–7. [\[CrossRef\]](#)
33. Alessandri, A.; Donnarumma, S.; Martelli, M.; Vignolo, S. Motion Control for Autonomous Navigation in Blue and Narrow Waters Using Switched Controllers. *J. Mar. Sci. Eng.* **2019**, *7*, 196. [\[CrossRef\]](#)
34. Donnarumma, S.; Figari, M.; Martelli, M.; Zacccone, R. Simulation of the Guidance and Control Systems for Underactuated Vessels. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer Nature: London, UK, 2020; Volume 11995, pp. 108–119. [\[CrossRef\]](#)
35. Alessandri, A.; Donnarumma, S.; Luria, G.; Martelli, M.; Vignolo, S.; Chiti, R.; Sebastiani, L. Dynamic positioning system of a vessel with conventional propulsion configuration: Modeling and simulation. Maritime Technology and Engineering. In Proceedings of the MARTECH 2014: 2nd International Conference on Maritime Technology and Engineering, Lisbon, Portugal, 15–17 October 2014; pp. 725–734.
36. Martelli, M.; Faggioni, N.; Zacccone, R. Development of a navigation support system by means of a synthetic scenario. Sustainable Development and Innovations in Marine Technologies. In Proceedings of the 18th International Congress of the International Maritime Association of the Mediterranean, IMAM 2019, Varna, Bulgaria, 9–11 September 2019; pp. 481–487.
37. Friedman, J.H.; Bentley, J.L.; Finkel, R.A. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw. (TOMS)* **1977**, *3*, 209–226. [\[CrossRef\]](#)
38. Donnarumma, S.; Figari, M.; Martelli, M.; Vignolo, S.; Viviani, M. Design and Validation of Dynamic Positioning for Marine Systems: A Case Study. *IEEE J. Ocean. Eng.* **2018**, *43*, 677–688. [\[CrossRef\]](#)