



Simulation of the Guidance and Control Systems for Underactuated Vessels

Silvia Donnarumma^(✉), Massimo Figari, Michele Martelli, and Raphael Zaccone

Department of Electrical, Electronics and Telecommunication Engineering and Naval Architecture (DITEN), University of Genoa, 16145 Genoa, Italy
donnarumma@dime.unige.it, {massimo.figari,michele.martelli, raphael.zaccone}@unige.it

Abstract. Intelligent and/or autonomous vehicle technologies are rapidly growing to meet the needs of marine safety and transport efficiency. One of the requirements to manage autonomous vehicles includes the integration between route planning and automatic motion control. In the authors' opinion, the latter could be sketched in three different layers: obstacle detection, planning and actuation. Moreover, the three layers should be able to interact in real-time. Dealing with such a challenging task, one of the best techniques to develop and test the logic is the use of the time-domain simulation. In the present work, a simulation model, integrating a path planning algorithm in the presence of obstacles with a track keeping controller, is developed. The path planning is based on a modified version of the Rapidly-exploring Random Tree (RRT*) algorithm. The track keeping is based on the Line-of-Sight (LOS) waypoints navigation for underactuated vessels. To achieve more reliable results, a detailed ship simulation model is used as a benchmark. Different scenarios and navigation modes are successfully tested, and the results are presented and analysed.

Keywords: Autonomous vessels · Simulation · Track keeping · Collision avoidance

1 Introduction

Modern ships are attracting attention by the research community owing to the increased on-board electronic devices and computational resources, likewise recently occurred for autonomous vehicles such as cars, planes, helicopters, and trains [1].

Technological breakthroughs arising from dynamic positioning applications have shown that it is possible to operate a vessel autonomously in a restricted area, under the supervision of on-board personnel, [2, 3]. Simultaneously, on-shore, the goals subsequently achieved with automotive applications have brought benefits to the driving comfort of vehicles. Similarly, the autonomous industry is now able to provide navigation support that can suggest the best instructions to the commander to cross crowded

navigation areas or to take charge of the management of the vehicle under his supervision. The possibility of integrating these solutions into the navigation system under standard conditions, thanks to the available technological support, would allow masters to have a system capable of solving instantly very complex optimization problems. Safe navigation and obstacle avoidance are some of the most relevant challenges on which research on autonomous ships is based.

Nowadays, attention on crew training and emergency procedures are milestones for reducing the risk of accidents at sea. Standards regulating navigation in crowded areas are reported in COLREGS [4], but the success of the manoeuvres depends on both environmental and human factors. The collision avoidance has always been a crucial topic in the marine scientific community, the work by Skjong [6] presents a differential formulation of the collision avoidance problem, together with an algorithm to compute the solution by numeric integration. Most of the marine applications are based on classic heuristic algorithms, such as genetic algorithms (GA) [7–9]. Hasegawa in [10] presents a different approach based on potential field methods. More recently, the rapidly-exploring random tree (RRT) algorithm, and its optimizing version RRT* have been studied into details. RRT algorithm has been introduced by LaValle & Kuffner [11], who analyze its performances on various applications. In particular, [12] compares RRT to dynamic programming in planning feasible (non-optimal) trajectories. The effectiveness of these algorithms is analyzed in [5], together with reference herein, and a new “ad hoc” path planning methodology is proposed. The core of the presented algorithm is indeed to use the computational power available in every ICT device to help the bridge operators in potentially dangerous situations, without taking any action, yet acting as a virtual bridge assistant that suggests the correct manoeuvre.

On the other hand, developing new control systems able to steer vessels under the supervision of the commander is the subject of marine engineering research since the early 1900s, [13]. As previously mentioned, the coupling of the strong results coming from the automotive industry together with the great results for station keeping, sparked the interest of the sector in the possibility of providing devices able to both suggest or actuate command directly. Such devices can limit stress in some short term demanding operations or, maybe even more, long term standard operations. For such a reason, guidance systems have been developed together with the increasing diffusion of inertial navigation systems [14]. As proposed in [15], and reference herein, guidance is, indeed, of crucial importance for autonomous ships and demands for the use of different motion models [16–19]. A well-known approach is based on selecting a sequence of waypoints and constructs the desired path as the sum of the successive straight lines and [20] provides solutions able to achieve smoothing transitions between waypoint by relaxing the turning problem. In [15, 21] authors show results from guidance and control systems designed to be robustly integrated into a complex controller orchestra able to steer an underactuated vessel from blue to narrow water.

This paper aims to present results from the methodologies implemented in order to improve ship safety, preventing grounding and collision and, at the same time, to support navigation. Indeed, the presented simulation platform was initially developed trying to solve two independent problems: the evaluation of the best evasive route and the development of a control system able to allow an under-actuated vehicle to maintain a predetermined trajectory.

In the present case study results from the integration of the, already implemented, collision avoidance together with the, already tested, path following are presented in a unique simulation platform. In particular, the presented work deals with a vessel, fitted with standard propulsion configuration (two propellers with rudders and a bow thruster). The simulator adopted for the integration with the whole controller is a multi-physics platform, able to represent the dynamics of a ship in three degrees of freedom, taking into account for the entire propulsion system together with automation. Detailed information of each sub-module can be found in [22, 23].

2 Simulation Platform

In this section, the motion controller structure is presented. In particular, Fig. 1 shows a flow-chart of the system. Analyzing the diagram from the top there are the inputs provided by the commander: the waypoints that identify the desired manoeuvre. At each time the collision avoidance system checks for possible obstacles. If no collision is possible, the waypoints are directly sent to the navigation system that translates the waypoints into heading requirements to ensure the vessel can regain course within a certain user-defined length, [5]. Otherwise, the system calculates an evasive manoeuvre that provides for the subsequent return into the course by generating new waypoints that are sent to the guidance system. On the bottom of the figure, the controller converts the course requirements signals into rudder angles and bow-thruster commands, in accordance with [15].

Then, the purpose of a marine automatic collision avoidance system is to detect the obstacles, plan and, perhaps, actuate the evasive manoeuvre. A detection system is able to identify the presence of obstacles in the environment and rapidly update the path by providing a set of waypoints to safely guide the ship through the target, following COLREG. The planned manoeuvre is represented in the following form:

$$P_k = (x_k, y_k), \quad k = 0, \dots, N \quad (1)$$

where P_k is the k^{th} waypoint, and $0, N$ represent the start and the target point, respectively. The motion planning system takes into account the manoeuvring capabilities of the ship, in order to provide a feasible evasive path.

The guidance system is based on the well-known *line-of-sight* algorithm, modified in accordance with [15]. Indeed, the proposed controller deals with 2 degrees of freedom and use propellers and rudders in autopilot mode with the possibility of also activating the bow thruster in a *drift-free track* condition. A proportional-integral-derivative (PID) controller has been synthesized similar to the one proposed in [21] fed with tracking error:

$$e_\psi = \psi - \psi_{des} \quad (2)$$

where ψ is the actual vessel heading and ψ_{des} is the desired heading.

Eventually, the switching or concurrent use of bow-thruster and rudders for course keeping is subject to a left-over speed law that allows propulsion to be used in high-efficiency conditions.

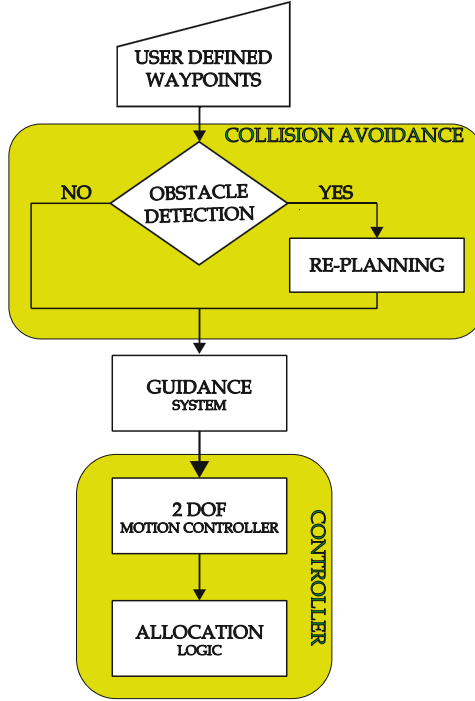


Fig. 1. Collision avoidance and controller structure flow-chart.

2.1 Ship Modeling

In this section, the several models used to identify the dynamic behaviour of all the main ship subsystem are reported.

2.1.1 Kinematics

Let $\eta := [x, y, \psi]^T \in \mathbb{R}^3$ be the array expressing the position and orientation of the vessel with respect to the Earth-Fixed Frame, and let $\mathbf{v} := [u, v, r]^T \in \mathbb{R}^3$ be the array containing the linear and angular velocity components with respect to the Body-Fixed basis. The kinematic relation between η and \mathbf{v} is given by the following expressions [20]:

$$\dot{\eta} = \mathbf{R}(\psi)\mathbf{v}, \quad \mathbf{R}(\psi) = \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3)$$

2.1.2 Dynamics

The notation $\boldsymbol{\tau} := [X, Y, N]^T \in \mathbb{R}^3$ is introduced for the array containing the components of longitudinal and lateral forces and their resultant moment. The ship motion equations are:

$$M\dot{\boldsymbol{v}} + C(\boldsymbol{v})\boldsymbol{v} + D_0\boldsymbol{v} + D(\boldsymbol{v})\boldsymbol{v} = \boldsymbol{\tau}_D + \boldsymbol{\tau}_E \tag{4}$$

Where M, C and D represent the mass-inertia and added mass, Coriolis and damping matrices respectively. $\boldsymbol{\tau}_D$ are the delivered forces and moments and $\boldsymbol{\tau}_E$ represent the environmental forces and moments [24], not taken in to account in this case.

The whole simulation platform representing the detailed system is shown in Fig. 2.

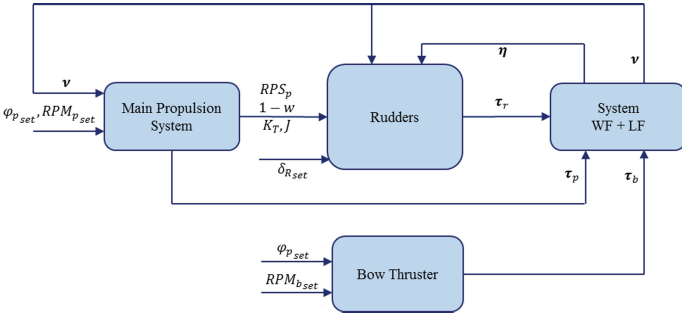


Fig. 2. Detailed simulation model

The main elements of the propulsion plant are two main engines, two gearboxes, two controllable pitch propellers and one controllable pitch bow-thruster. The ship is equipped with two independent shaft lines, the main elements of the propulsion plant are two main engines, two gearboxes, and two controllable pitch propellers, the simulation blocks that identify one shaft line is reported in Fig. 3. One controllable pitch bow-thruster and two rudders constitute the manoeuvring devices.

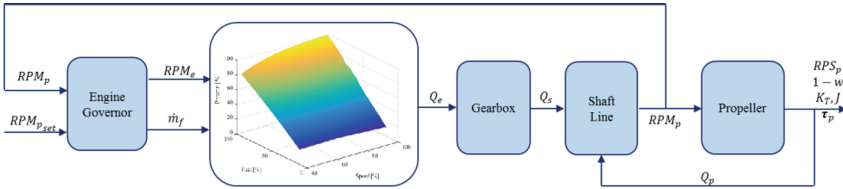


Fig. 3. Propulsion plant model

2.1.3 Engine

The engine governor computes the fuel flow requirement on the basis of the propeller speed error. It is a PI controller with the following law:

$$\dot{m}_f^r = K_{PRPM} e_{RPM} + \int_0^t K_{IRPM} e_{RPM}(\xi) d\xi \tag{5}$$

Where $e_{RPM} = \frac{100}{N_{MAX}}(N_e^r - RN_s^r)$ is the error from the setpoint coming from the controller and K_{PRPM} and K_{IRPM} are the constant coefficients of the regulator. Then, the fuel flow is saturated in accordance with the maximum engine torque limits:

$$\dot{m}_f = \begin{cases} \dot{m}_f^r & \dot{m}_f^r < \dot{m}_f^{(max)} \\ \dot{m}_f^{(max)} & otherwise \end{cases} \quad (6)$$

2.1.4 Shaft Line

Equation (7) describes the shaft line dynamics:

$$\frac{d\omega}{dt} = \frac{Q_E(t) - Q_P(t) - Q_f(t)}{I_e + I_g + I_s + I_p} \quad (7)$$

where $I_e + I_g + I_s + I_p$ represent the engine, gear, shaft and propeller rotational inertia; Q_E , Q_P and Q_f are the engine, propeller and frictional torque, respectively. In order to catch the asymmetric behaviour of a twin-screw ship, the dynamics of the two shaft lines have been implemented separately.

2.1.5 Propeller

In literature, several numerical methods based, for instance, on the potential approach or R.A.N.S.E. solvers, have been proposed to predict propeller hydrodynamic loads. Unfortunately, due to their long computational time, these methods are not suitable to be applied in the context of a time-domain simulator like the one described in the present work. Therefore, the hydrodynamic forces, both for propeller and bow thruster, have been evaluated through a quasi-steady-state methodology based on the propeller open water tests. These tests provide an open water diagram, which allows the evaluation of the thrust coefficient K_T and torque coefficient K_Q .

For controllable pitch propellers, these coefficients depend on the advance coefficient J and from the blade angle φ :

$$K_T = \frac{T}{\rho n^2 D^4}, \quad K_Q = \frac{Q}{\rho n^2 D^5} \quad (8)$$

2.1.6 Rudder

For the same reasons as before, also in this case the rudder forces have been evaluated through a quasi-static-methodology, using lift and drag coefficients C_L and C_D . In addition, the complex interaction between rudder and propeller has been taken into account using the approach described in [25].

2.2 RRT* Algorithm

This section presents a brief overview of the RRT algorithm application to ship motion control. RRT [11] is a space-exploring algorithm that generates a tree of feasible paths, that explore the space trying to reach all the feasible configurations from a starting point. RRT can take into account holonomic and non-holonomic constraints.

The algorithm builds a tree that is composed of a set of nodes and a set of edges E , growing it for a fixed number of iterations. First, the algorithm generates a new random position in the domain, then finds the node that is nearest to it.

A new node is then generated in accordance with the constraints, moving by a fixed step from the nearest node in the direction of the new random position.

The RRT algorithm does not perform any optimization neither computes any cost or return function, i.e. it does not provide any ranking criterion for the solutions, in favour of the computation speed. A heuristic variant of the RRT, named RRT*, is based on the RRT algorithm, has some additional steps in order to heuristically drive the tree growth to minimize a cost function. In particular, two separate actions, both aiming to locally drive the growth through optimality, take place before the new node is added, to select his parent and child nodes in accordance to the cost function.

The selection of a proper cost function allows, thus controlling the trajectories generated by the algorithm [5]. Note that, unlike in the RRT, an increase in the number of iterations in RRT* leads to better solutions.

In the proposed approach, constraints are set to generate collision free trajectories respecting the kinematic limits of the ship [26]. In particular, the trajectories are required to keep a minimum distance from the obstacles and the other ships: the threshold distance needs to be properly selected, and should also take into account the uncertainties coming from the obstacle localization system, as well as the possible overshoots from the waypoints due to the motion control and actuation. Further details and a potential implementation of COLREGs can be found in [27].

2.3 Motion Control

As briefly introduced in Sect. 1, the motion control is based on two independent controllers that provide for both the vessel heading and the speed. In such a way, the sway and the yaw are controlled, based on the algorithm known as line-of-sight approach, with the *drift-free track* while the surge is indirectly controlled through the *speed-pilot*. The integrated controller orchestra is detailed in [15].

In particular, both the controller structures are PID type and the coefficient synthesis has been carried out by using optimization algorithms based on linear matrix inequalities (LMI) technique, [21].

3 Simulation Results

This section compares results from the simulations of the proposed navigation system with and without the collision avoidance system. Such simulations also take into account the presence of moderate environmental disturbances by the action of a wind coming from the east with an average speed of 10 knots. In Figs. 4 and 6 the blue curves represent the trajectory of the ship navigating autonomously via the blue markers, while the red curve represents an obstacle that heads over to the trajectory of the ship at a constant speed. In Figs. 5 and 7 the blue curve represents the distance of the ship from the obstacle and the red dashed curve is the safety distance. All the quantities shown are dimensionless in ship lengths.

As can be seen in Fig. 4, the autonomous navigation system crosses all the required waypoints without taking into account the presence of other units in the surrounding environment. In Fig. 5, the distance from the obstacle overlaps the limits reaching zero. Figure 6 shows the effects of collision avoidance. In fact, the system replaces the user-defined requirements with the appropriate commands able to avoid the collision and, actually, by handling the path planning of the vehicle during the obstacle overing. Eventually, user-defined waypoints back into the navigation system after the obstacle has been overcome. Indeed, the ship remains to its original course. In this case, it can be seen in Fig. 7 that the distance from the obstacle remains, for the entire manoeuvre, within the safety domain.

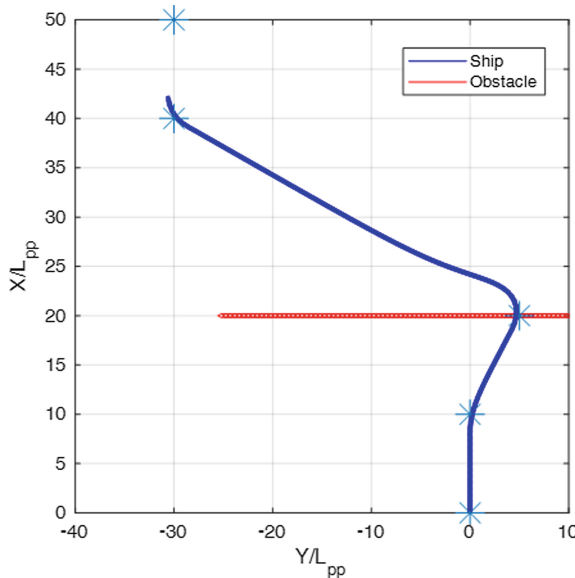


Fig. 4. Vessel and obstacle paths without collision avoidance. (Color figure online)

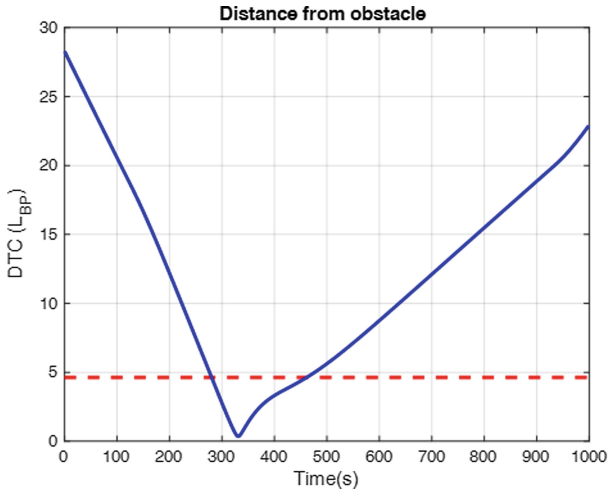


Fig. 5. Vessel distance from the obstacle when the collision avoidance is not active. (Color figure online)

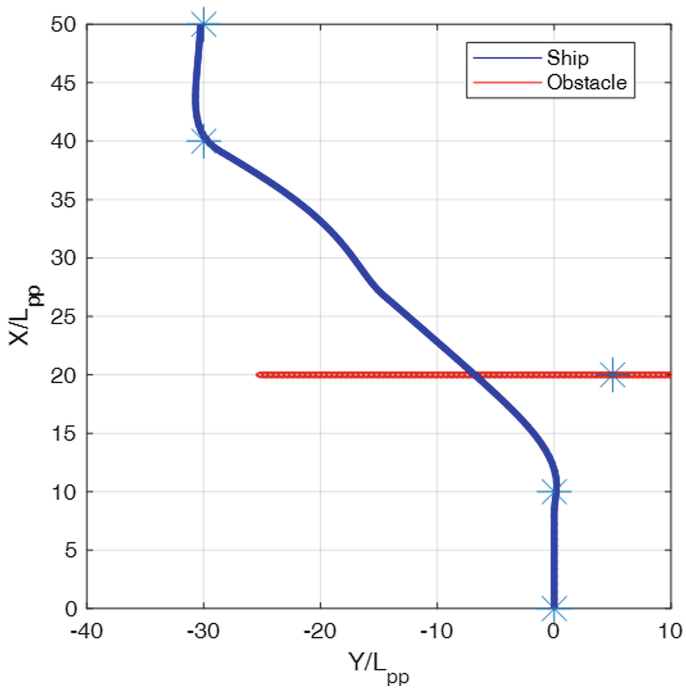


Fig. 6. Vessel and obstacle paths when collision avoidance is active. (Color figure online)

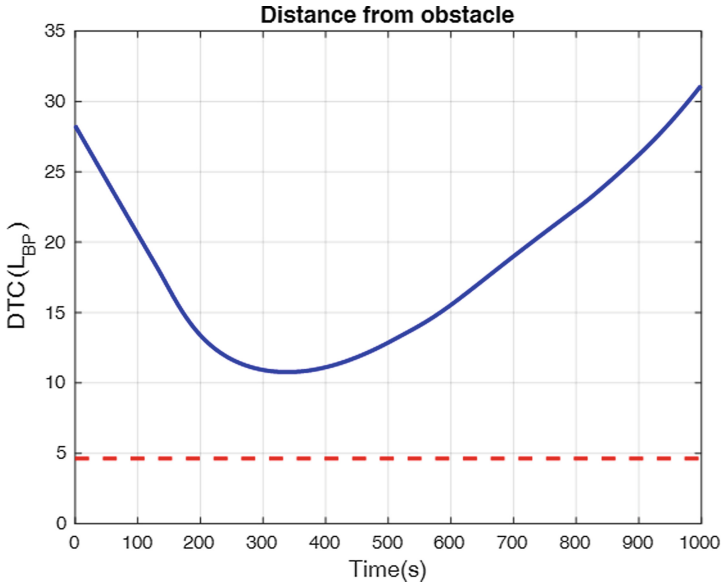


Fig. 7. Vessel distance from the obstacle when the collision avoidance is active. (Color figure online)

4 Conclusions

The integration of a collision avoidance logic together with a guidance system able to allow safe autonomous navigation has been described, and results are critically compared. At this stage the proposed avoidance methodology can deal in real-time in case of complex scenarios, taking into account for environmental disturbances, albeit limited. Optimization problems are suitable in terms of computational time for the relatively slow ship dynamics and long distances. The novelty with respect to references can be evaluated in the automatic return to the original course, without human actions. Moreover, the robustness of each sub-system of the proposed structure has been validated in both stand-alone and orchestral mode by giving a large multi-purpose domain to the system. Indeed, the user can act in each part of the structure by using the outputs of each block as suggestions, and without diverging the algorithms. Further research will concern the validation of the model in more complex scenarios and the integration in the algorithm of information about the environmental disturbances will be crucial part of the path replanning.

References

1. Stateczny, A., Burdziakowski, P.: Universal autonomous control and management system for multipurpose unmanned surface vessel. *Pol. Marit. Res.* **26**, 30–39 (2018)
2. Fay, H.: *Dynamic Positioning Systems: Principles, Design, and Applications*. OPHRY (1990)

3. Sorensen, A.J.: A survey of dynamic positioning control systems. *Annu. Rev. Control* **35**, 123–136 (2011)
4. COLREGS - International Regulations for Preventing Collisions at Sea
5. Zaccone, R., Martelli, M.: A random sampling based algorithm for ship path planning with obstacles. In: *Proceedings of the International Ship Control Systems Symposium (iSCSS)*, 2–4 October 2018. <http://doi.org/10.24868/issn.2631-8741.2018.018>
6. Skjong, R., Mjelde, K.M.: Optimal evasive manoeuvre for a ship in an environment of fixed installations and other ships. *Model. Ident. Control* **3**, 211–222 (1982)
7. Ito, M., Zhng, F., Yoshida, N.: Collision avoidance control of the ship with a genetic algorithm. In: *Proceedings of the 1999 IEEE International Conference on Control Applications*, vol. 2, pp. 1791–1796. IEEE (1999)
8. Smierzchalski, R., Michalewicz, Z.: Modelling of ship trajectory in collision situations by an evolutionary algorithm. *IEEE Trans. Evol. Comput.* **4**(3), 227–241 (2000)
9. Alvarez, A., Caiti, A., Onken, R.: Evolutionary path planning for autonomous underwater vehicles in a variable ocean. *IEEE J. Oceanic Eng.* **29**(2), 418–429 (2004)
10. Hasegawa, K., Fukuto, J., Miyake, R., Yamazaki, M.: An intelligent ship handling simulator with automatic collision avoidance function of target ships. In: *Proceedings of INSLC 2017* (2012)
11. LaValle, S.M., Kuffner Jr., J.J.: Rapidly-exploring random trees: progress and prospects. In: *4th Workshop on the Algorithmic Foundations of Robotics; Algorithmic and Computational Robotics, New Directions*, Hanover, NH, pp. 293–308 (2000)
12. LaValle, S.M.: From dynamic programming to RRTs: algorithmic design of feasible trajectories. In: Bicchi, A., Prattichizzo, D., Christensen, H.I. (eds.) *Control Problems in Robotics*, pp. 19–37. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36224-X_2
13. “Now – The Automatic Pilot” *Popular Science Monthly*, February 1930, p. 22
14. Draper, C.: Control, navigation, and guidance. *IEEE Control Syst. Mag.* **1**, 4–17 (1981)
15. 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. JSME* **7**(6). <https://doi.org/10.3390/jmse7060196>
16. Rauch, H.E.: Autonomous control reconfiguration. *IEEE Control Syst. Mag.* **15**, 37–48 (1995)
17. Lekkas, A., Fossen, T.: Integral LOS path following for curved paths based on a monotone cubic Hermite spline parametrization. *IEEE Trans. Control Syst. Technol.* **22**, 2287–2301 (2014)
18. Fossen, T., Pettersen, K.: On uniform semiglobal exponential stability (USGES) of proportional line-of-sight guidance laws. *Automatica* **50**, 2912–2917 (2014)
19. Paliotta, C., Lefeber, E., Pettersen, K.Y., Pinto, J., Costa, M.: de Figueiredo Borges de Sousa, J.T.: Trajectory tracking and path following for underactuated marine vehicles. *IEEE Trans. Control Syst. Technol.* **27**(4), 1423–1437 (2019)
20. Fossen, T.: *Marine Control Systems*. Marine Cybernetics, Trondheim (2002)
21. Alessandri, A., et al.: System control design of autopilot and speed pilot for a patrol vessel by using LMIs. In: Soares, C.G., Dejhalla, R., Pavletić, D. (eds.) *Proceedings 16th International Congress of the International Maritime Association of the Mediterranean*. CRC Press, pp. 577–584 (2015)
22. Alessandri, A., et al.: Dynamic positioning system of a vessel with conventional propulsion configuration: Modeling and simulation (2015) *Maritime Technology and Engineering - Proceedings of MARTECH 2014: 2nd International Conference on Maritime Technology and Engineering*, pp. 725–734
23. Donnarumma, S., Martelli, M., Vignolo, S.: Numerical models for ship dynamic positioning *MARINE 2015 - Computational Methods in Marine Engineering VI*, pp. 1078–1088 (2015)
24. Martelli, M.: *Marine Propulsion Simulation*, pp. 1–104. De Gruyter open (2015)

25. Martelli, M., Viviani, M., Altosole, M., Figari, M., Vignolo, S.: Numerical modelling of propulsion, control and ship motions in 6 degrees of freedom. In: Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment, vol. 228, pp. 373–397, November 2014
26. Zaccone, R., Martelli, M.: A collision avoidance algorithm for ship guidance applications. *J. Marine Eng. Technol.* **19**, 62–75 (2019)
27. Zaccone, R., Martelli, M., Figari, M.: A colreg-compliant ship collision avoidance algorithm. In: 18th European Control Conference, ECC 2019, art. no. 8796207, pp. 2530–2535 (2019)