Fuzzy domain and meta-heuristic algorithm-based collision avoidance control for ships: Experimental validation in virtual and real environment

Remzi Fiskin[a,*], Oguz Atik[b], Hakki Kisi[c], Efendi Nasibov[d] and Tor Arne Johansen[e]

[a]*Department of Marine Transportation Engineering, Fatsa Faculty of Marine Sciences, Ordu University, Ordu, Turkey, e-mail: remzifiskin@odu.edu.tr, phone: +905074742450*
[b]*Department of Marine Transportation Engineering, Maritime Faculty, Dokuz Eylül University, İzmir, Turkey, e-mail: oguz.atik@deu.edu.tr*
[c]*Department of Marine Transportation Engineering, Maritime Faculty, Dokuz Eylül University, İzmir, Turkey, e-mail: hakki.kisi@deu.edu.tr*
[d]*Department of Computer Science, Faculty of Science, Dokuz Eylül University, İzmir, Turkey,*
e-mail: efendi.nasibov@deu.edu.tr
[e]*Department of Engineering Cybernetics, Faculty of Information Technology and Electrical Engineering, Norwegian University of Science and Technology, Trondheim, Norway, e-mail: tor.arne.johansen@ntnu.no*
[*]*Corresponding author*

Abstract

Collisions at sea are frequently caused by human-related factors, such as; manoeuvre timing mistakes, risk assessment failures and deficiencies in strategies for collision avoidance. These factors reveal the importance of the automation systems in providing safety of navigation. Thus, a decision support system was developed in this study that can be a reference to the ship operators in the implementation of the collision avoidance action, in case of an encounter situation involving risk of collision. Both qualitative and quantitative methods were conducted in the study. In the qualitative research process, the variable constraints in the mathematical model and the inputs of the scenarios implemented in experiments were determined based on the findings obtained from experts' interviews. In the quantitative research process, the problem-solution was reached with the developed algorithm (ColAv_GA), which is formed based on the Genetic Algorithm and Fuzzy Logic. The developed algorithm was validated in a virtual environment using a bridge simulator, and in a real environment with an autonomous surface vehicle (ASV), with satisfactory results. The output of this research is expected to contribute to the safety of navigation. The developed algorithm can be used as a collision avoidance sub-module for autonomous ships and ASV.

Keywords: Genetic Algorithm, Fuzzy Logic, Collision Avoidance, Ship

1. Introduction

In dynamic environments such as aviation and maritime transportation, collision is one of the most important problems, which brings negative impacts to human lives and the environment (Lazarowska, 2015a; S. Li et al., 2019b, 2019a). In order to overcome such a problem, it is necessary to place special emphasis on training and establish effective procedures. However, the most important factor in such a situation is the experience of the navigators, who perform the precautionary action. The navigators practically perform this action based on experience rather than the training they received, and the procedures to be followed. However, even an experienced navigator can make a wrong assessment and cause an accident (Perera et al., 2015). For this reason, limiting subjective decisions taken by navigators in navigation, and supporting them with a decision support system, will undoubtedly reduce the risk of collision and contribute to increasing the safety of navigation (Perera et al., 2011; Tsou and Hsueh, 2010).

Navigation is a complex process, as it requires continuous analysis and monitoring of large amounts of data. Increasing marine traffic, especially, makes navigation more difficult and complicated for navigators (Lazarowska, 2012). Navigation includes the interaction of factors such as; environment, ship and human, dynamic and static data, certain and uncertain information, quantitative calculation and qualitative reasoning (Tsou and Hsueh, 2010). For this reason, making a navigation assessment, considering these factors, is critical for the safety of navigation. Improper and inadequate assessment may lead to accidents so, this assessment process should be supported (Lazarowska, 2012).

The collision avoidance action is a multi-criteria problem. While applying this action, there should be a balance between the safety of navigation and navigation cost (Smierzchalski and Michalewicz, 1998). In other words, the collision avoidance process should not only include collision risk assessment and avoidance action but also consider optimizing the amount of deviation distance from the original route (Su et al., 2012).

This research consists of two main parts, as qualitative and quantitative research process, for the solution of the collision avoidance problem at sea. As a result of the findings obtained using the interview method in the qualitative research process, the inputs of the scenario tests and the variable constraints in the mathematical model created for problem-solving were determined.

In the quantitative research process, the designed algorithm structure was coded with C# programming language on Microsoft Visual Studio 2017 platform. Problem solution was reached with this algorithm (called ColAv_GA), which is based on GA and fuzzy logic approaches. The solutions generated by the algorithm were tested in a virtual environment with a bridge simulator (TRANSAS NTPRO 5000) in Turkey, and in a real environment with an autonomous surface vehicle (ASV) in Norway.

The remaining sections of the paper are organized as follows: Section 2 introduces the literature review and describes the originality of the research, followed by Section 3, which provides the methodology and the model explanation of the proposed method. Section 4 presents experimental test results, and Section 5 provides a discussion and a comparison with other models proposed by various authors, followed by the conclusions, including final evaluation of the results, and recommendations for the further studies in Section 6.

2. Literature Review and Originality

Conditions such as manoeuvre timing errors, collision risk assessment failures and deficiencies in strategies necessary for collision avoidance, which are human based, are the main indicators that cause collision (Tsou and Hsueh, 2010). Automation and decision support systems become important in terms of eliminating or minimizing these human-based errors. From this point of view, the psychological and physical burden on the navigator will decrease with the decision support system. This will reduce the risk of collision and contribute to the safety of navigation. On the other hand, the system can be integrated as a collision avoidance sub-module in autonomous ships and unmanned surface vehicles.

Collison causes not only financial loss but also injuries and large environmental pollution. Therefore, collision avoidance is one of the issues that have been discussed in recent years, especially with the emerging concept of autonomous and unmanned ships (Tsou, 2019). Researchers have proposed various methods, techniques and models with different approaches on ship encounter situations and collision avoidance route problem at sea which have become an important issue, especially over the past decade. It is possible to divide these studies into 4 groups according to approach types; deterministic, artificial intelligence (AI), hybrid systems and simulation. Fiskin et al. (2018) reviewed these approaches in detail with a systematic approach.

Deterministic methods always produce exact and identical results for every execution (Akkoyunlu and Engin, 2011). Such methods follow a strict and precise execution process. It is commonly stated that the most traditional and classical algorithms have a deterministic structure (Yang et al., 2010), and nowadays they lose their effectiveness and have difficulty in explaining the complexity in real-time decision making processes (Büyükyazıcı and Taşar, 2011). In related field, deterministic approaches (e.g., Fiskin et al., 2019; Szlapczynski, 2008; Szlapczyński, 2007; Tam & Bucknall, 2010; Yavin et al., 1998) include certain analytical definitions to provide exact results related to navigation and route problem. Deterministic methods have an advantage over AI-based methods because of providing exact results.

AI methods basically consist of fuzzy logic, heuristic and meta-heuristic algorithms, artificial neural networks (ANN) and similar methods. The studies by Grinyak and Devyatisil’nyi (2016), Lee et al. (2015), Perera et al. (2011, 2010), Su et al. (2012) and Li et al. (2019) are some important studies based on fuzzy logic. On the other hand, as an example of the AI approach, Cheng et al. (2007), Hao et al. (2007), Kang et al. (2018), Lazarowska (2015b, 2014) and Tsou et al. (2010) proposed heuristic and meta-heuristic based methods, and Lisowski (2000) and Simsir et al. (2014) introduced ANN-based methods to solve the problem. The most obvious difference between these types of algorithms from other approaches is that they can produce near-optimal results in a short time for complex problems with their learning capacity and high computing ability.

Since the 1970s, researches have been basically carried out the application of standard heuristic and meta-heuristic algorithms (Kaveh and Ilchi Ghazaan, 2018). However, thereafter, hybrid algorithms emerged with an idea that the uniform algorithm is limited in problem-solving and that combining it with another algorithm may be more effective and feasible, especially in solving large-scale real-life problems (Blum and Roli, 2008). Hybrid algorithms are algorithms created by combining at least two different algorithms to provide a more efficient execution process. In related literature, Cheng and Liu (2007) and Perera et al. (2015) conducted studies to solve the problem with a hybrid approach. Cheng and Liu (2007) combined genetic algorithm and simulated annealing algorithm and Perera et al. (2015) combined fuzzy logic and ANN to create hybrid models.

A simulation method is defined as combining methods and applications in order to imitate the behaviour of real systems using computers with the appropriate software. In other words, it is the process of designing and creating computer-aided models for a better understanding of the behaviours shown by real systems that form under certain conditions (Kelton et al., 2003). Models created with the simulation technique may not always offer optimum results. The performance of a system is measured by examining all the scenarios with different variables. For this reason, a simulation method is a tool that tries to optimize a system considering various scenarios rather than directly optimizing it (Esmer, 2009). Studies conducted by Liu et al. (2007) and Johansen et al. (2016) are based on the simulation method. Liu et al. (2007) proposed a model based on multi-agent simulation and Johansen et al. (2016) introduced an approach based on the predictive control model.

As a result, this approach uses an AI method based on fuzzy logic and a meta-heuristic algorithm. With the applications and achievements below, this research is thought to have an original contribution considering the related literature;

- Determination of the size of the ship safety domain of the own ship (OS) with the fuzzy inference system (FIS) created by considering factors determined as a result of expert interviews.
- Unlike similar studies, in this approach, in accordance with International Regulations for Preventing Collisions at Sea (COLREG) Rule 16, the position, where the collision avoidance manoeuvre is to be applied, is determined by the system user considering the distance of the target ship (TS) (called as Mx_{toTS}) (i.e., action range to the TS).
- Unlike similar studies, in this approach, in accordance with COLREG Rule 8 (d), manoeuvring to return the original route is made after the TS is considered as clear.
- Unlike similar studies, in this approach, offering the option of producing results according to the optimization goal (shortest path or Mx_{toTS}) of the system user.
- Determining the constraints used in the algorithm structure and sample scenario inputs as a result of the findings obtained from expert interviews conducted within the scope of the qualitative research process.

- Controlling the validity and reliability, and testing the applicability of the system with scenario tests were performed in a virtual environment with bridge simulator and in a real environment with ASV.
- Capability to produce results with course alteration or speed change manoeuvre.

In case of a risk of collision between two ships at sea, the give-way ship needs to alter her course or speed to eliminate the risk as per COLREG. In such cases, the Automatic Radar Plotting Aid (ARPA) radar is a critical device to get assistance. ARPA reports whether there is a potential collision risk between the two ships, but does not provide any optimal routes to avoid collision. Collision avoidance action, performed by the navigator subjectively based on experience/competence and ARPA supported, will be carried out through a decision support system. Ultimately, this study is considered to be important due to the following possible circumstances;

- The system contributes to the reduction of human errors arising from subjective judgments.
- The system also contributes to the enhancement of navigational safety by reducing the psychological and physical burden on the decision-maker in decision making.
- The system can be a reference in decision making for Vessel Traffic Services (VTS) operators and navigators.
- The system aims to provide the optimal collision avoidance route. It hereby will be possible to save fuel and time.
- The system will contribute to increasing the safe ship control automation.
- The system can be integrated as a collision avoidance sub-module in autonomous ships and unmanned surface vehicles.

3. Methodology

The method of the study basically consists of two main parts. In the qualitative research process, a total of 10 interviews were organized with ship masters and officers to determine the variable constraints in the mathematical model, the factors affecting the ship domain and the inputs of the scenario tests. In the quantitative research process, the results obtained with the solution algorithm (ColAv_GA), created based on GA and fuzzy logic approaches, were tested in a virtual environment with the bridge simulator, and in a real environment with ASV.

3.1. Conceptual Framework of the Study

The conceptual framework of the study mainly consists of a total of 3 steps as shown in Fig. 1. At the first step, a preliminary survey was conducted by taking the expert judgments and recommendations regarding the research subject and the related literature were examined. In the second step, exploratory researches were conducted. In this research process, the findings arising from the qualitative research methods, using the interview technique, were analysed in order to determine the variable constraints, factors affecting ship safety domain and inputs of scenario tests. Then, by creating a GA and fuzzy logic-based algorithm structure, the initial results regarding the solution of the problem were obtained. At the last stage, the developed system was tested in a virtual and real environment for validity and reliability.

3.2. Problem Definition

The most important cause of marine accidents is the human factor (Bowo and Furusho, 2019). Operators tend to make mistakes by nature, even if they have higher education and experience. Intelligent decision support and guidance systems, therefore, have an important place in terms of navigational safety and useful to guide the operators (Lee et al., 2019; Perera et al., 2015). Encounter situations at sea occur in two sub-environmental situations: coastal areas and open sea. Although there are many external systems (VTS, separation line, etc.) that provide support in coastal areas, there is no navigation system at open sea to support navigation in order to avoid collision. In encounter situations involving collision risk at open sea, navigators decide the action they take by getting support from the bridge devices. ARPA Radar, which is the most commonly utilized bridge device in encounter situations, is a critical device in this regard. Although the “trial manoeuvre” feature gives an idea about the movement of the ships in time, it cannot automatically calculate the optimal collision avoidance action to be performed. For this reason, this study focuses on the optimal collision avoidance route planning at open sea or non-congested waters.

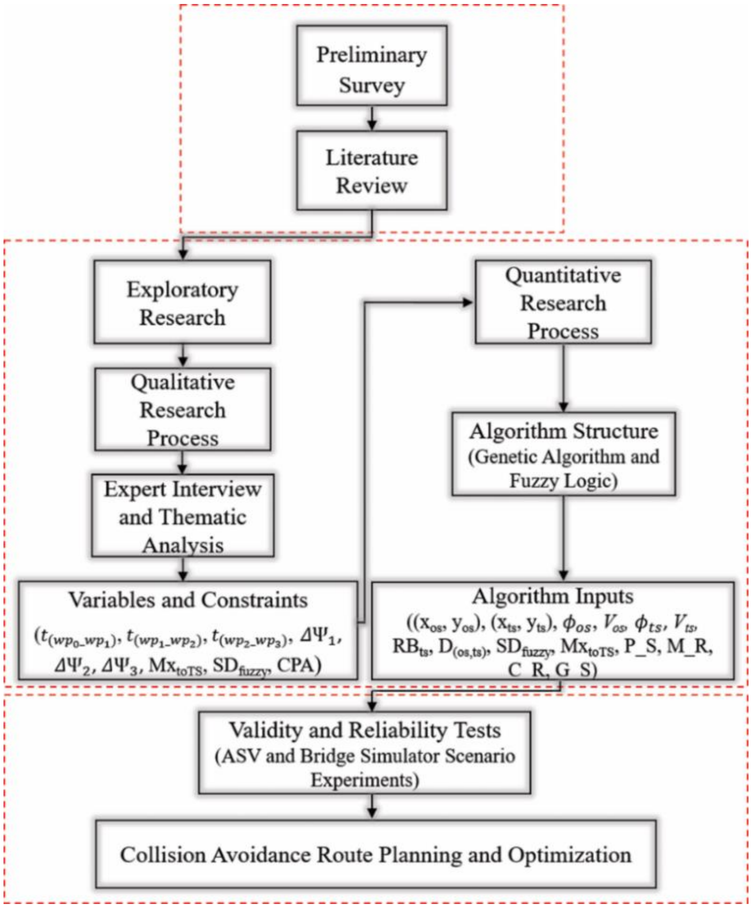


Fig. 1. Conceptual framework of the study.

In case of a collision risk at sea, actions to be taken by the ship officers are regulated by the COLREG rules. The ship, which is defined as a give-way vessel according to the rules, must alter her course, speed or both in order to eliminate collision risk. COLREG classifies the ships as a give-way vessel and a stand-on vessel (in case of encounter) and constitutes regulations for one-to-one encounter situations involving only two vessels. For this reason, problems arise in complex traffic conditions. In such cases, in practice, the navigators communicate with each other and they try to conduct their manoeuvring decision considering COLREG rules. This study, therefore, focused on one-to-one encounter situation as in many similar studies (Brcko and Švetak, 2013; Candeloro et al., 2017; Fiskin et al., 2019; Hongdan et al., 2015b, 2015a, 2014; Kang et al., 2018; Mostefa, 2014; Naeem et al., 2012; Perera et al., 2015; Szlapczynski, 2015; Tsou et al., 2010; Wei et al., 2015; Xu, 2014; Xu et al., 2014). It is possible to divide the collision avoidance manoeuvre into four phases according to the different purposes it has (Li and Ma, 2016; Tsou et al., 2010).

i. *Initial navigation phase:* At this phase, the closest point of approach (CPA) of the nearby obstacles (e.g., other ships and buoys) are tracked. In case of an obstacle that violates the ship domain, i.e., minimum CPA, the necessary alarms are kept in order to give an alert.
ii. *Warning phase:* In the event that an obstacle being tracked violates the ship domain, if the obstacle is a ship, the type of encounter and the roles of the ships are determined according to the COLREG. If the OS is the give-way vessel, the vessel obliges to make the necessary manoeuvre to eliminate the risk of collision. TS as a stand-on vessel, on the other hand, is responsible for maintaining the current motion (course and speed).
iii. *Collision avoidance manoeuvre phase:* Collision avoidance manoeuvre is performed in this phase. The manoeuvre to be performed should be neither too small for the TS to perceive nor large to get too far from the original route. In this phase, the aim is to eliminate the risk of collision by preventing the violation of ship domain, in principle.
iv. *Returning to the original route:* After making sure that the collision avoidance manoeuvre is successfully completed and the target is clear, returning to the original route is performed.

3.3. Mathematical Model and Algorithm Structure

3.3.1. Fitness Function Model

This study, theoretically, aims to obtain the minimum collision avoidance route length. More specifically, it is aimed to minimize the total length of route leg formed from the position *(wp_1)* where the ship starts the manoeuvre to avoid collision to the position *(wp_2)* where it manoeuvres to return to the original route, and the route leg formed from the position *(wp_2)* where it manoeuvres to return to the original route to the position *(wp_3)* where it enters the original route. The fitness function determined for this purpose is defined by equation 1.

$$f(x) = \min \sum_{i=1}^{n-1} m_i \tag{1}$$

where *f(x)* denotes the fitness function, *n* is the number of the wp, m_i is the distance passed in the *i.* leg of the collision avoidance route.

As shown in Fig. 2, the collision avoidance route consists of a total of 3 wp (i.e., two route leg). Therefore, since n = 3, the fitness function can be expressed in another way by equation 2.

$$f(x) = \min\{m_{(wp_1\_wp_2)} + m_{(wp_2\_wp_3)}\} \tag{2}$$

where $m_{(wp_1\_wp_2)}$ is the distance passed after the collision avoidance manoeuvre (the length of the first leg of the collision avoidance route), $m_{(wp_2\_wp_3)}$ is the distance passed after returning manoeuvre to the original route (the length of the second leg of the collision avoidance route) representing with equation 3 and equation 4, respectively.

$$m_{(wp_1\_wp_2)} = t_{(wp_1\_wp_2)} V_{os}(t) \tag{3}$$

$$m_{(wp_2\_wp_3)} = t_{(wp_2\_wp_3)} V_{os}(t) \tag{4}$$

where $t_{(wp_1\_wp_2)}$ is the proceeding time in the first leg of the collision avoidance route, $t_{(wp_2\_wp_3)}$ is the proceeding time in the second leg of the collision avoidance route, V_{os} is the speed over ground of the OS.

Equation 5-8 represent the motion model of the OS and TS using the kinematic model. The distance between the ships ($D_{(os\_ts)}(t)$) is calculated with equation 9 using the 2-dimensional Euclidean distance formula.

$$x_{os}(t) = x_{os}(t-1) + Sin(\phi_{os})V_{os}(t) \quad (5)$$
$$y_{os}(t) = y_{os}(t-1) + Cos(\phi_{os})V_{os}(t) \quad (6)$$
$$x_{ts}(t) = x_{ts}(t-1) + Sin(\phi_{ts})V_{ts}(t) \quad (7)$$
$$y_{ts}(t) = y_{ts}(t-1) + Cos(\phi_{ts})V_{ts}(t) \quad (8)$$
$$D_{(os\_ts)}(t) = \sqrt{(x_{os}(t) - x_{ts}(t))^2 + (y_{os}(t) - y_{ts}(t))^2} \quad (9)$$

where $x_{os}(t)$ is the x-axis value of the position of the OS at time *t*, $y_{os}(t)$ is the y-axis value of the position of the OS at time *t*, $x_{ts}(t)$ is the x-axis value of the position of the TS at time *t*, $y_{ts}(t)$ is the y-axis value of the position of the TS at time *t*, ϕ_{os} is the course of the OS, ϕ_{ts} is the course of the TS, V_{ts} is the speed of the TS.

The relative velocity between ships ($V_{(os\_ts)}$) is calculated by equation 10 using the cosine theorem.

$$V_{(os\_ts)} = \sqrt{V_{os}^2 + V_{ts}^2 - 2V_{os}V_{ts}cos\phi_{(os\_ts)}} \quad (10)$$

Where $\phi_{(os\_ts)}$ is the difference between courses of the OS and TS and defined by equation 11.

$$\phi_{(os\_ts)} = \phi_{os} - \phi_{ts} \quad (11)$$

3.3.2. Decision Variables

As illustrated in Fig. 2, the algorithm structure contains a total of 4 decision variables for the course change manoeuvre. These decision variables are as follows;

$t_{(wp_0\_wp_1)}$: Proceeding time in minutes from the initial position (*wp₀*) to position of course change to avoid collision (*wp₁*).

$\Delta\Psi_1$: Course change to be applied at *wp₁* to avoid collision.

$t_{(wp_1\_wp_2)}$: Proceeding time in minutes from wp_1 to position of course change to return the original route (wp_2).

$\Delta\Psi_2$: Course change to be applied at wp_2 to return the original route.

3.3.3. Constraints

In optimization problems, the best result of the fitness function is searched in the range of constraints determined for decision variables. In this study, the constraints determined for the decision variables are defined by equation 12-16, respectively.

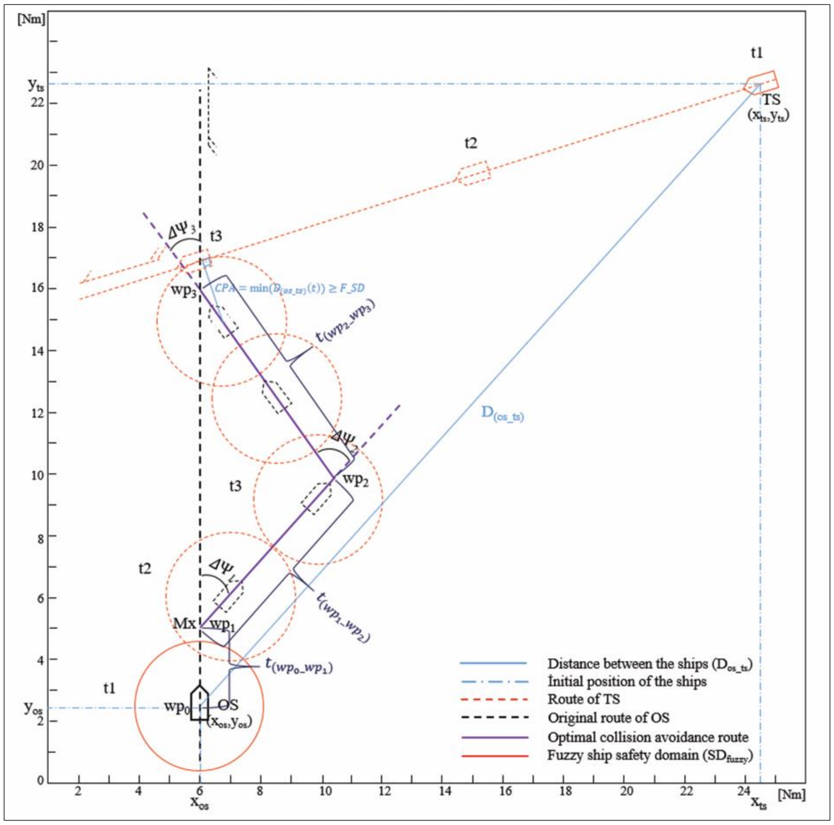


Fig. 2. Decision variables.

$$10 \leq t_{(wp_0\_wp_1)} \leq 63 \;\&\; t_{(wp_0\_wp_1)} < tCPA \tag{12}$$

$$5 \leq \Delta\Psi_1 \leq 63 \tag{13}$$

$$5 \leq t_{(wp_1\_wp_2)} \leq 31 \tag{14}$$

$$-63 \leq \Delta\Psi_2 \leq -10 \tag{15}$$

$$CPA = min(D_{os,ts}(t)) = SD_{fuzzy} \tag{16}$$

$t_{(wp_0\_wp_1)}$ ($\in$ [10,63] in minute): Unlike existing models in the literature, the value of this variable can be obtained according to two different approaches: (1) making a course change to avoid collision before the distance to the TS set by the system user (Mx_{toTS}), and (2) by GA without determining any distance to the TS. These approaches, which are provided as two options, have advantages and disadvantages comparing them within each other. The interviewed experts noted that the former one is more suitable for real-life practices. The experts also added that they make the manoeuvre at a certain distance (mostly 8-10 Nm) to the TS. When this distance, called Mx_{toTS}, is provided to the system, GA is not looking for a value for this variable since it is obvious at which position the OS turns. This provides an advantage in two perspectives: being more suitable for real-life practices and decreasing execution time to reach the optimal solution. This approach also meets the "*give-way vessel shall, so far as possible, take early and substantial action to keep well clear*" rule in COLREG Rule 16. In the second approach, GA provides this variable. In this approach, ships can be very close to each other at the manoeuvring time since the distance to which the manoeuvre is to be applied has not been determined. However, it is considered to be applicable for low speeds or ships with high manoeuvrability. The advantage of this approach is that the optimal collision avoidance route length is shorter than generated by the other approach. The interviewed experts stated that the proceeding time between 10 minutes to 1 hour until the collision avoidance manoeuvre is sufficient. Considering this, the variable is represented by a 6-bit in the binary number system [$111111_2=63_{10}$]. This variable has to be less than time to the closest point of approach (tCPA).

This variable can be obtained according to two different approaches by Mx_{toTS} or by GA which are described in detail above. When this variable is determined by GA, it has been realised that there may be more convergence between ships than it should be. Interviewed experts stated that a collision avoidance manoeuvre should be performed at a specified distance from the target ship. For this reason, the Mx_{toTS} option that allows the user to determine this distance has been introduced. Determining this parameter with GA ensures that the total length of the collision avoidance trajectory is shorter, but more convergence between ships involves risks in terms of navigational safety. Although determination with Mx_{toTS} increases the total length of the collision avoidance trajectory, it is more applicable in practice.

$\Delta\Psi_1$ ($\in$ [5,63] in degree): Course alteration manoeuvre should be made in accordance with COLREG Rule 16 which states that *"it will be so obvious that another vessel can easily perceive"*. According to the findings obtained as a result of the question addressed to the experts in order to reveal how this rule is applied in practice, the constraint of this variable was determined between 5^o to 60^o. Similar to the previous one, the constraint was determined to be 63 instead of 60, as it is represented by a 6-bit. The larger this variable, the harder the ship has to turn. As the angle of manoeuvre increases, the loss of speed in turns increases and the distance to be covered by the ship can be different from the planned accordingly. Therefore, the lower this value, the closer the ship can sail to her planned trajectory.

$t_{(wp_1\_wp_2)}$ ($\in$ [5,31] in minute): COLREG Rule 8(d) dictates that *"action taken to avoid a collision with another vessel shall be such as to result in passing at a safe distance"*. In accordance with this rule, the OS should wait for the return to the original route until the TS becomes clear. As a result of expert interviews, it was revealed that when the TS is boarded, it is accepted as clear. The constraint of this variable was also determined between 5 minutes to half an hour. This constraint of this variable is represented by a 5-bit. Since the maximum value of a 5-bit is 31 in the binary number system, the constraint interval is set to 31 instead of 30 [11111_2=31_{10}].

$\Delta\Psi_2$ (∈ [-10,-63] in degree): This constraint of this variable is represented by a 6-bit between 10° to 60° similar to the constraint of the second variable. Where minus (-) denotes turning to the reverse side, i.e., to the port side. The value produced here is added to the $\Delta\Psi_1$, resulting in a course manoeuvre degree for returning to the original route.

Finally, the last constraint is the distance between the ships when they are closest ($min(D_{os,ts}(t))$, called CPA, which should be greater than or equal to the ship domain (SD_{fuzzy}) of the ship. In order to obtain the minimum collision avoidance route length, it must be equal.

3.3.4. Assumptions

Some assumptions were considered to simplify the complexity of the problem before the algorithm structure is formed. These assumptions are as follows;

i. Collision risk assessment is based on whether the ship domain is violated.

ii. As a stand-on vessel, the TS keeps its motion by maintaining speed and course.

iii. Navigational data of ships (speed, route, etc.) is obtained from any system. In practice, this data is obtained through devices such as Automatic identification system (AIS), ARPA, Electronic Chart Display and Information System (ECDIS).

iv. The ships are considered to comply with the COLREG rules and act according to the rules.

v. Ship movements are calculated with a kinematic model approach.

vi. Loss of time and speed while manoeuvring is ignored.

vii. The ships sail in calm water condition but not in the sea state.

3.3.5. Genetic Algorithm Structure

In this study, the binary coding system [0,1] was used to encode the chromosomes. In this system, the length of chromosomes is equal to the total number of genes of each decision variable. Each chromosome represents a possible solution according to the fitness function. With this coding, there are a total of $2^{23} = 8388608$ chromosomes and possible solutions represented by each chromosome. Considering the studies in the literature, the number of chromosomes in the population was determined to be 50. The gene arrays of the chromosomes were randomly sorted, and the number of individuals was kept constant in each generation. After calculating the fitness value of each chromosome in the population, the roulette wheel and elitism method were used for parent selection. After the selected individuals were randomly matched, a single-point crossover operator was applied for the next generation production with the possibility of 0.7 crossover rate. Then, a bitwise mutation operator was applied to the individuals in the new generation with the probability of 0.03 mutation rate. The algorithm was terminated by determining the maximum number of iterations as 100 iterations. GA parameters, the algorithm flowchart and the pseudo code used in this study are shown in Table 1, Fig. 3 and Table A.2, respectively.

Table 1
GA parameters used in the algorithm.

Parameter	Value	Operator	Formation
Genetic representation	([0],[1])	Coding	Binary system coding
Initial population (P_S)	50 (constant)	Population creation	Coding each individual and keeping a result
Selection process	-	Parent selection	Roulette wheel, elitism
Crossover rate (C_R)	0.7	Crossover	Single-point crossover
Mutation rate (M_R)	0.03	Mutation	Bitwise
Termination criterion (G_S)	100	Fitness calculation	Minimum route length

3.3.6. Collision Risk Assessment

Collision risk assessment is a fundamental concept in sea navigation. There are a total of 3 main methods to conduct a collision risk assessment in practice: methods based on traffic flow theory, ship safety domain and methods based on distance to closest point of approach (dCPA) and tCPA (Xu and Wang, 2014; Zhang and Meng, 2020). In this study, collision risk assessment was carried out with the ship safety domain.

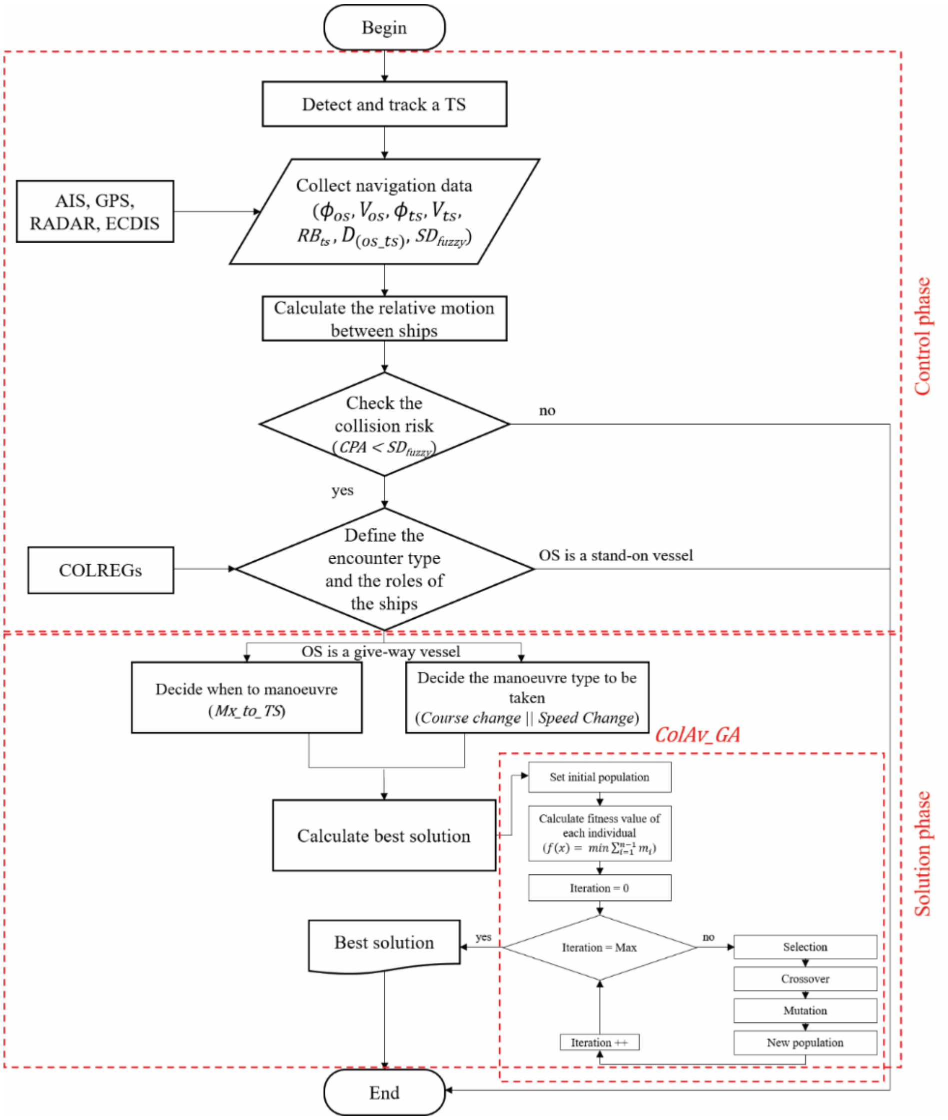


Fig. 3. Flowchart of the algorithm.

The ship safety domain (hereinafter called as ship domain), an area that surrounds the ship, keeps the ship away from other surface objects (i.e., other ships, objects, etc.) (Goodwin, 1975). Each ship on navigation sets a ship domain. Violation of this area by any stationary or moving object is considered the risk of collision. Other objects must be kept out from this area to eliminate the risk of collision.

There are many different approaches applied in determining the ship domain in the literature. These methods are classified as follows: Empirical, knowledge-based, analytical and probabilistic (Fiskin et al., 2020; Szlapczynski and Szlapczynska, 2017). In these methods, the shape of the ship domain is focused differently. The circle is a shape of ship domain that researchers often work on (Davis et al., 1980; Jingsong et al., 1993; Zhu et al., 2001). Moreover, elliptical (Coldwell, 1983), hexagon (Lisowski et al., 2000), irregular shape (Pietrzykowski, 2008), quadrangle (Dinh and Im, 2016) and polygonal (Fiskin et al., 2020) are other proposed ship domain types. In this study, a circle-shaped ship domain developed with a knowledge-based approach was used to assess the collision risk. The circle-shaped ship domain is the most widely used form of ship domain in practice.

The ship domain is typically determined by masters based on their experience and expertise. In addition, companies also determine the minimum required distance between a target and a ship. This minimum distance is included in the company navigation policy. Fuzzy Inference System (FIS) was used in this study for determining the ship domain size, considering that it would be important to determine which factor is important to the size of the ship domain which is determined generally based on experience. Mamdani type FIS method, fuzzy set and fuzzy rule base were created by using multiple-input single-output (MISO) IF–THEN rules.

In creating a fuzzy ship domain (SD_{fuzzy}), Triangular and Trapezoid membership functions for Fuzzification, the Sum method as the Aggregation method, and the Centroid method as the Defuzzification method were employed. The system created with MATLAB R2019b fuzzy logic tool was defined over 8 linguistic input variables and 1 linguistic output variable. Input variables were determined as a result of expert interviews as follows*: navigator experience (K), weather condition (S), ship size (B), ship speed (H), manoeuvrability (M), traffic state (T), day-night (G), visibility (V).* The output variable was determined as *the radius of the ship domain* (SD_{fuzzy})*.* This can be expressed briefly as $SD_{fuzzy} = [K, S, B, H, M, T, G, V]$ with an eight-dimensional (n = 8) input vector. A total of 231 fuzzy rules were created with all input variables. The model structure of the FIS including fuzzy numbers, type of membership functions and universal size defined for each linguistic variable is similar to the model proposed by Yardımcı et al. (2019) and Fiskin (2019). The structure of the FIS is illustrated in Fig. 4.

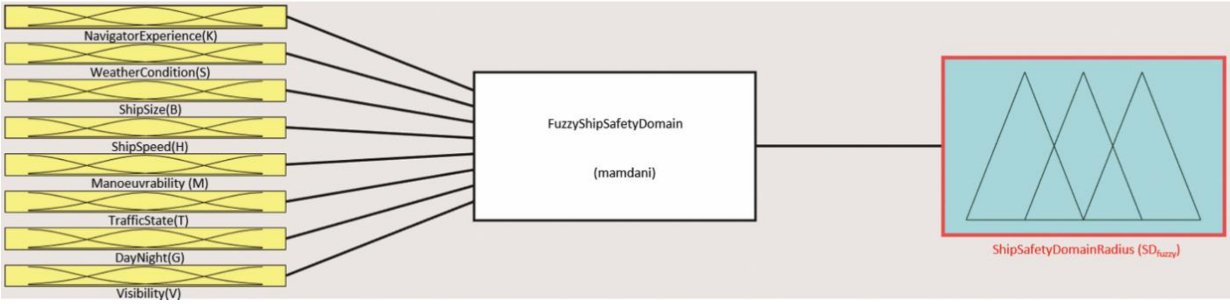


Fig. 4. The structure of the proposed FIS.

The order of importance was made for input variables based on expert interviews. According to this order of importance, weight coefficients are defined for fuzzy rules in which the variables take place. The weight coefficients of fuzzy rules determine their effects on the fuzzy system. The coefficients of fuzzy rules are determined as follows. The ratio of the number of statements taken by the experts for each input variable in the total number of statements was determined. *Ship size (B)* has the highest rate with 25.93 (category a). Each of the variables *ship speed (H), weather condition (S)* and *traffic state (T)* have a ratio of 14.82 (category b). Each of the variables *manoeuvrability (M), navigator experience (K), day-night (G)* and *visibility (V)* have a ratio of 7.41 (category c). Since each fuzzy rule includes two input variables, the calculation of the weight coefficients of the variables according to the categories is shown in table 2.

Table 2
Defining the weight coefficients of fuzzy rules.

Category	Average of the ratios		Weight coefficients
a+b	= (25.93+14.82)/2 = 20.38	Normalization ➡	0.79
a+c	= (25.93+7.41)/2 = 16.70		0.64
b+b	= (14.82+14.82)/2 = 14.82		0.57
b+c	= (14.82+7.41)/2 = 11.12		0.43
c+c	= (7.41+7.41)/2 = 7.41		0.29

4. Experimental Test Results

Experimental tests were applied to test the validity, reliability and applicability of the developed system. In this context, experimental tests were carried out in a virtual environment with the bridge simulator (in Maritime Faculty, Dokuz Eylül University (DEU), İzmir, Turkey), and in a real environment (in Trondheim Fjord, Trondheim, Norway) with ASV in cooperation with Norwegian University of Science and Technology, Centre for Autonomous Marine Operations and Systems (NTNU AMOS) and Maritime Robotics AS (MR). In these experiments, various scenarios were tested considering different ship encounter types. Ship encounter types are defined in COLREG rules as head-on, crossing and overtaking. Head-on indicates that two ships are approaching in reciprocal or nearly reciprocal directions and mainly refer to Rule 14. Crossing is that two ships are on cross direction, potentially involving the collision risk and mainly refer to Rule 15. Overtaking states that one ship is approaching with another ship in an angle larger than 22.5 degrees abaft her beam and mainly refer to Rule 13. In this paper, only a head-on situation experiment is presented for each experiment to demonstrate the system execution because of the limitation of the paper size.

4.1. Virtual Environment Experiment: Bridge Simulator Test

Scenario inputs for this experiment were set as SD_{fuzzy} 2 Nm, V_{os} 14 knots, ϕ_{os} 000°, V_{ts} 15 knots, ϕ_{ts} 184°, RB_{ts} 002°, $D_{(os\_ts)}$ 31 Nm and Mx_{toTS} 8 Nm as shown in Table 3.

Table 3
Virtual environment experiment scenario inputs.

	SD_{fuzzy} [Nm]	V [knot]	$\boldsymbol{\phi}$ [°]	$\boldsymbol{RB_{ts}}$ [°]	$\boldsymbol{D_{(os\_ts)}}$ [Nm]	Mx_{toTS} [Nm]
OS	2	14	000	-	-	8
TS	-	15	184	002	31	-

In this scenario, the CPA was calculated by the algorithm as 0.04 Nm (tCPA: 64.2', dCPA: 14.98' Nm). The value of the CPA must be greater than or equal to the ship domain in order to eliminate the risk of collision and ensure a safe passage ($CPA \geq SD_{fuzzy}$). In this case, since the ship domain size is determined as 2 Nm, it is necessary to make a collision avoidance manoeuvre assuming that there is a risk of collision between the ships. On the other hand, the CPA value was also calculated as 0.04 Nm by the bridge simulator. This shows that the developed algorithm calculates the CPA correctly.

The algorithm was run 100 times with a computer with Intel Core i7-7700 2.80 GHz processor, 16GB RAM, 64-bit operating system and reached the solution in periods ranging from 5.99 seconds to 6.73 seconds (mean: 6.35 seconds) as shown in Fig. 5. This demonstrates that the algorithm is applicable to real-time applications. The program produced identical results in every execution.

It was revealed that GA parameters do not affect the convergence to the best solution, because the identical solutions were returned in every execution. However, it was found out that the parameters only affect the speed of the algorithm to reach a solution which is illustrated in Fig. 5. As a result, different parameter settings can affect the computational time of the algorithm rather than the solution.

The solutions produced by the algorithm for head-on situation scenario are shown in Table 4. The values in the top row in the table are optimal values. The solution provided by the algorithm says that *“after proceeding $48^{\prime}$, turn 32^{o} to starboard, then, proceed on the new route for $16.8^{\prime}$ and turn 62^{o} to the opposite side (i.e., to port), then, proceed on the new route for $17.8^{\prime}$ and turn 30^{o} to starboard”*. When the OS moves according to this command, the length of the collision avoidance route is 8.07 Nm.

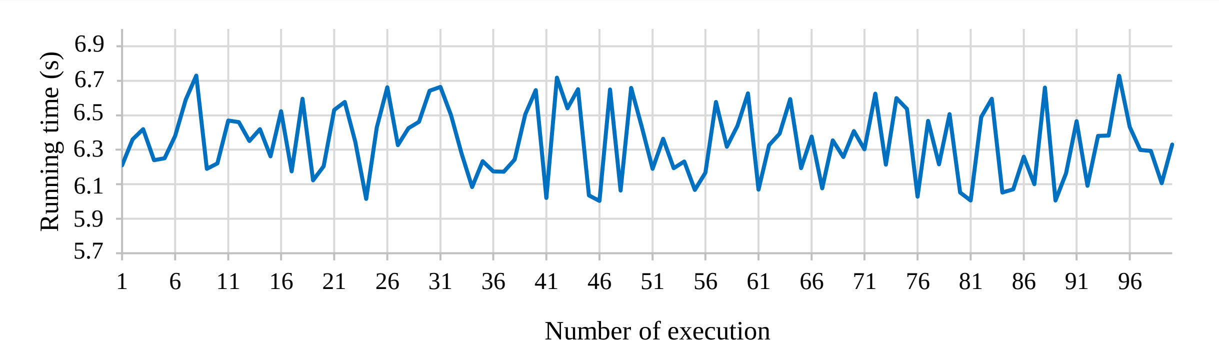
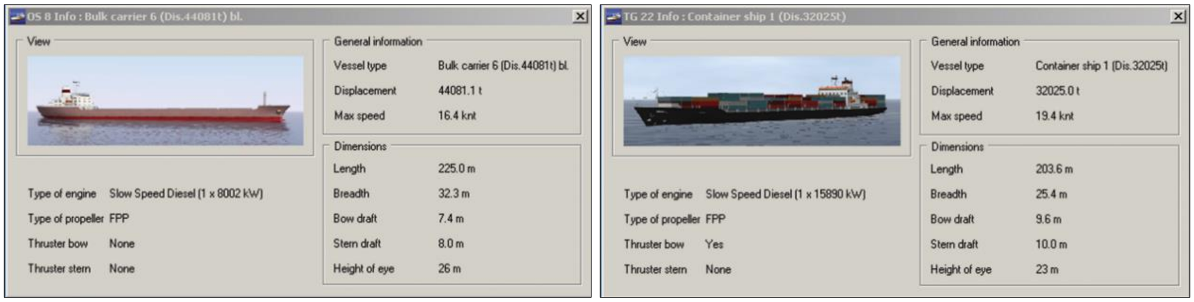


Fig. 5. Virtual environment experiment: running time for every execution.

Table 4
Virtual environment experiment: solutions produced by the algorithm.

$t_{(wp_0\_wp_1)}$ [m]	$\Delta\Psi_1$ [°]	$t_{(wp_1\_wp_2)}$ [m]	$\Delta\Psi_2$ [°]	$t_{(wp_2\_wp_3)}$ [m]	$\Delta\Psi_3$ [°]	$m_{(wp_1\_wp_2)} + m_{(wp_2\_wp_3)}$ [Nm]
48	**32**	**16.8**	**[-]62**	**17.81**	**30**	**8.07**
48	34	16.8	[-]64	18.79	30	8.3
48	40	15.6	[-]70	20.05	30	8.32
48	32	16.8	[-]60	18.96	28	8.34
48	48	14.4	[-]78	21.4	30	8.35
.	.	.	.	.	.	.
.	.	.	.	.	.	.

At this stage, a bridge simulator scenario test with the same scenario input was conducted to validate the solution provided by the algorithm. The scenario design of the bridge simulator was formed according to the sample scenario inputs. In this scenario, the OS and TS were chosen as a bulk carrier with a displacement of 44081(t) and a container ship with a displacement of 32025(t), respectively, as illustrated in Fig. 6.



OS 8 Info : Bulk carrier 6 (Dis.44081t) bl.

View

General information
Vessel type Bulk carrier 6 (Dis.44081t) bl.
Displacement 44081.1 t
Max speed 16.4 knt

Dimensions
Length 225.0 m
Breadth 32.3 m
Bow draft 7.4 m
Stern draft 8.0 m
Height of eye 26 m

Type of engine Slow Speed Diesel (1 x 8002 kW)
Type of propeller FPP
Thruster bow None
Thruster stern None

TG 22 Info : Container ship 1 (Dis.32025t)

View

General information
Vessel type Container ship 1 (Dis.32025t)
Displacement 32025.0 t
Max speed 19.4 knt

Dimensions
Length 203.6 m
Breadth 25.4 m
Bow draft 9.6 m
Stern draft 10.0 m
Height of eye 23 m

Type of engine Slow Speed Diesel (1 x 15890 kW)
Type of propeller FPP
Thruster bow Yes
Thruster stern None

Fig. 6. Virtual environment experiment: ships used in the experiment.

Fig. 7 shows the positions of the ships on the chart at the beginning of the scenario. According to the scenario inputs, the TS was positioned as relative bearing 002° and the distance is 31 Nm.

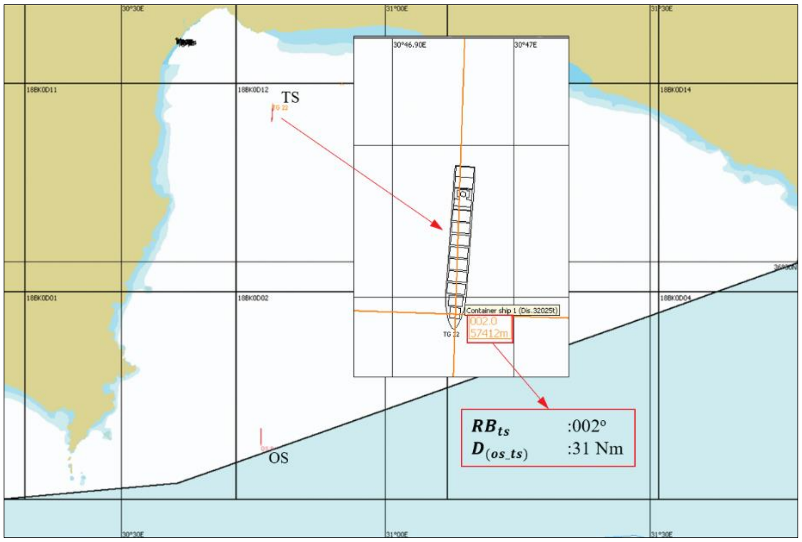


Fig. 7. Virtual environment experiment: location of the ships on the chart.

Fig. 8 and Fig. 9 show the stepwise comparative images of the bridge simulator and the developed program based on the developed algorithm. On the left side of the figures shows the chart screen of the bridge simulator, in the middle of the figures, shows the radar screen of the bridge simulator and on the right side of the figures shows the ship movement simulation based on the solution provided by the developed algorithm. In the chart screen, red lines denote trajectories of the ships. In the program screen, purple line and green lines denote the optimal route of the OS and other potential routes that are not optimal, respectively. Moreover, the green circle on the radar screen and the red circle on the program screen are the ship domain of the OS. As shown in Fig. 8, the OS proceeded for 48' ($t_{(wp_0\_wp_1)}$) according to the optimal result provided by the algorithm and reached the position (wp_1) where it would make the collision avoidance manoeuvre ($\Delta\Psi_1$). In this position, the OS turned 32° ($\Delta\Psi_1$) to starboard 8 Nm away from the TS (Mx_{toTS}) to make the collision avoidance manoeuvre. After the manoeuvre, the time $t_{(wp_1\_wp_2)}$, which is the time to proceed on the first leg of the collision avoidance route, has started. As shown in Fig. 9, the OS reached the position (wp_2) by proceeding 16.8' ($t_{(wp_1\_wp_2)}$) where it would turn to her original route. In this position, it turned (-)62° ($\Delta\Psi_2$) to the opposite side (i.e., port) to turn back to the original route. Then, the OS proceeded 17.8' ($t_{(wp_2\_wp_3)}$) and turned 30° ($\Delta\Psi_3$) to starboard as suggested by the algorithm. Finally, it entered her original route and finalized the collision avoidance action.

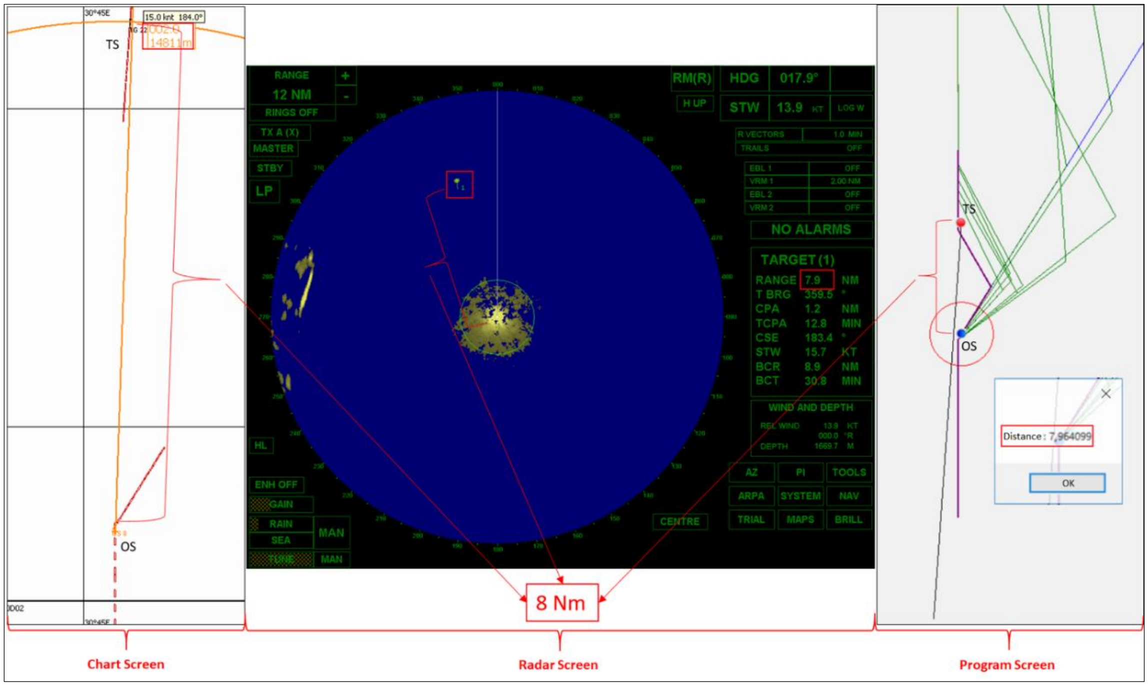


Fig. 8. Virtual environment experiment: stepwise comparison of the bridge simulator and the developed program (step 1).

On the control console of the bridge simulator, there is a feature that records and shows the trajectory of the ships. This control screen allows the distance covered by the ships to be measured. Fig. 10 shows the trajectory tracks of the ships that appeared in this scenario. The first leg of the collision avoidance route was measured as 3.9 Nm and the second leg as 4.17 Nm, and it turned out to be 8.07 Nm in total. This value is equal to the route length calculated by the algorithm. This demonstrates that the algorithm calculates correctly and properly. In order to accurately measure the performance of the developed algorithm, taking into account the loss of speed occurring in the turns, the simulator was paused when the ship reached the turning positions and was placed on the starting point of the next leg.

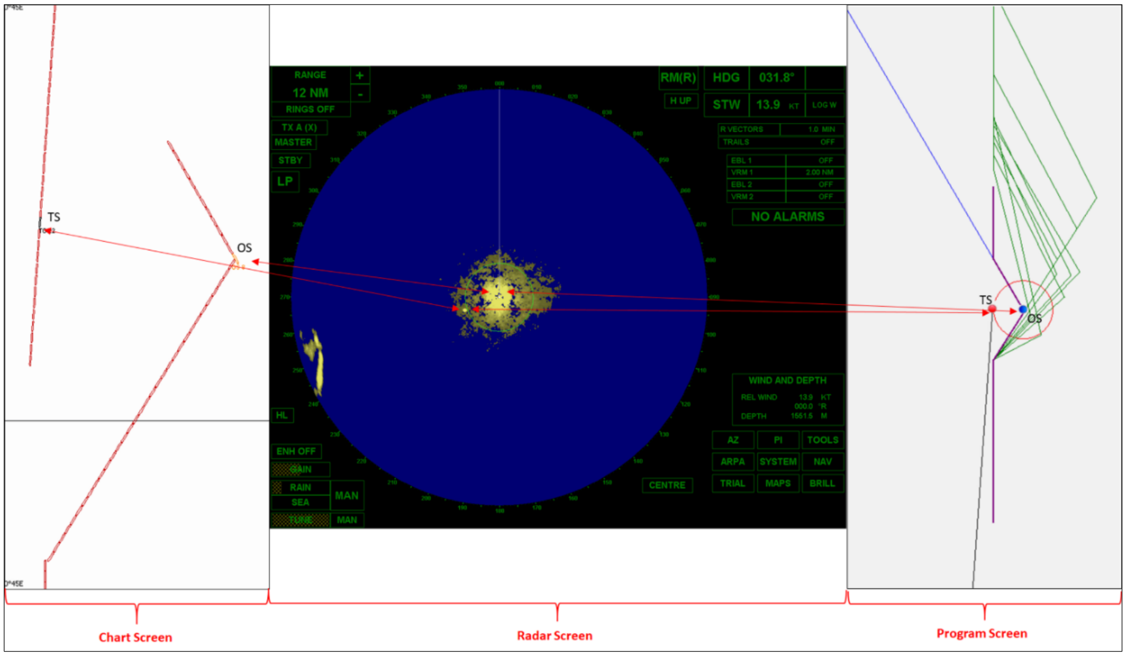
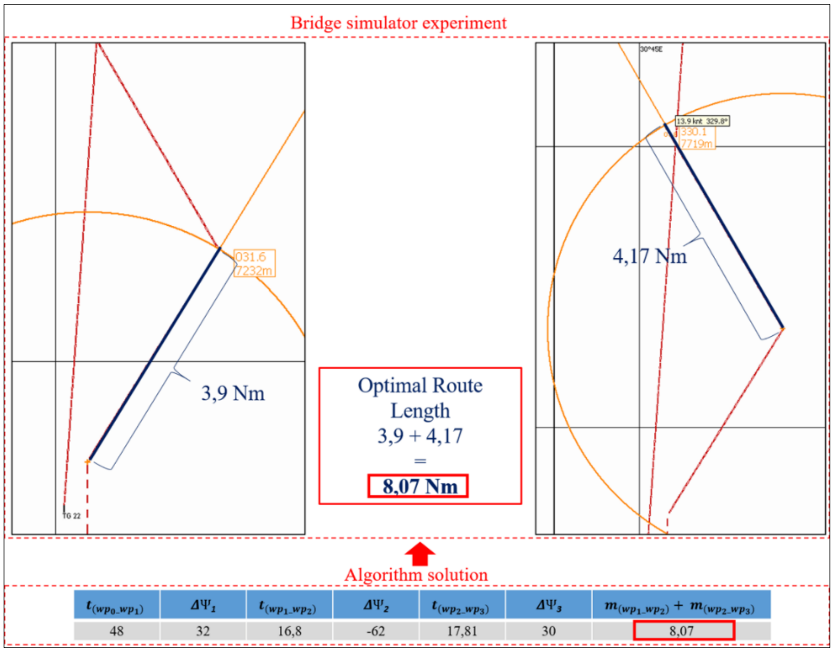


Fig. 9. Virtual environment experiment: stepwise comparison of the bridge simulator and the developed program (step 2).



$t_{(wp_0\_wp_1)}$	$\Delta\Psi_1$	$t_{(wp_1\_wp_2)}$	$\Delta\Psi_2$	$t_{(wp_2\_wp_3)}$	$\Delta\Psi_3$	$m_{(wp_1\_wp_2)} + m_{(wp_2\_wp_3)}$
48	32	16,8	-62	17,81	30	8,07

Fig. 10. Virtual environment experiment: bridge simulator optimal collision avoidance route length.

Fig. 11 shows the instantaneous distance graph between ships depending on time. It is revealed that the distance between the ships when they are closest to each other (i.e., CPA) is equal to the ship domain and the distance between the ships are increasing gradually after being closest.

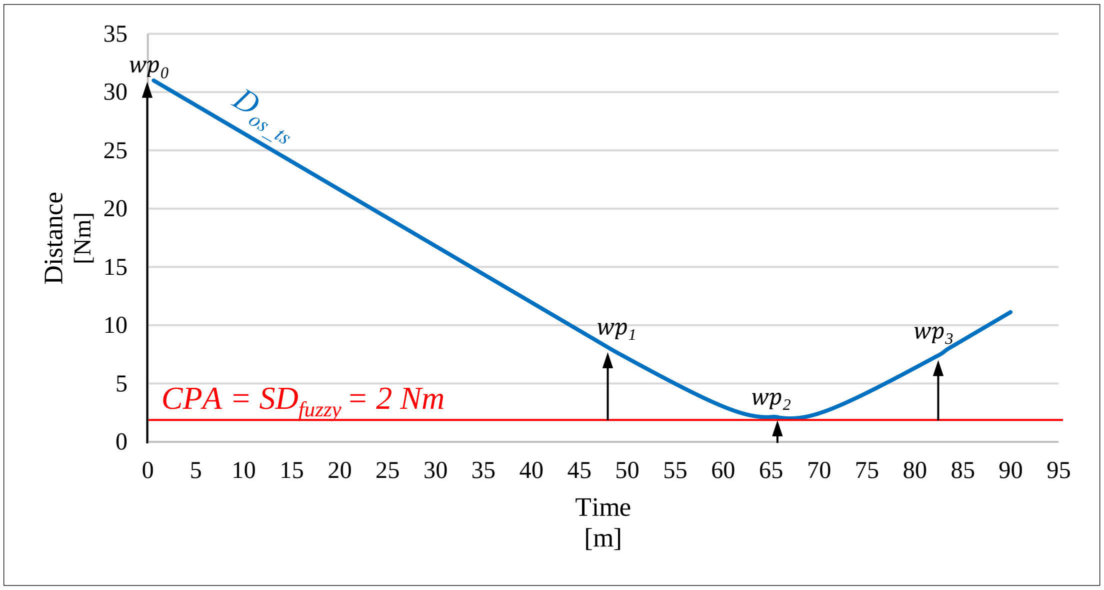


Fig. 11. Virtual environment experiment: instantaneous distance between ships.

Finally, Fig. 12 shows the values of the linguistic input variables defined for the FIS created for the SD_{fuzzy} calculation and the value of the output variable obtained as a result of these input variables. Each row in the figure represents one fuzzy rule. As shown in the figure, the values of linguistic input variables were defined as *[K, S, B, H, M, T, G, V] => [6,2,225,14,0.8,0.2,0.1,8]*. FIS calculated the ship domain radius as 2 Nm considering these inputs. This value was defined for the collision avoidance system as the ship domain of the OS in a virtual environment experiment.

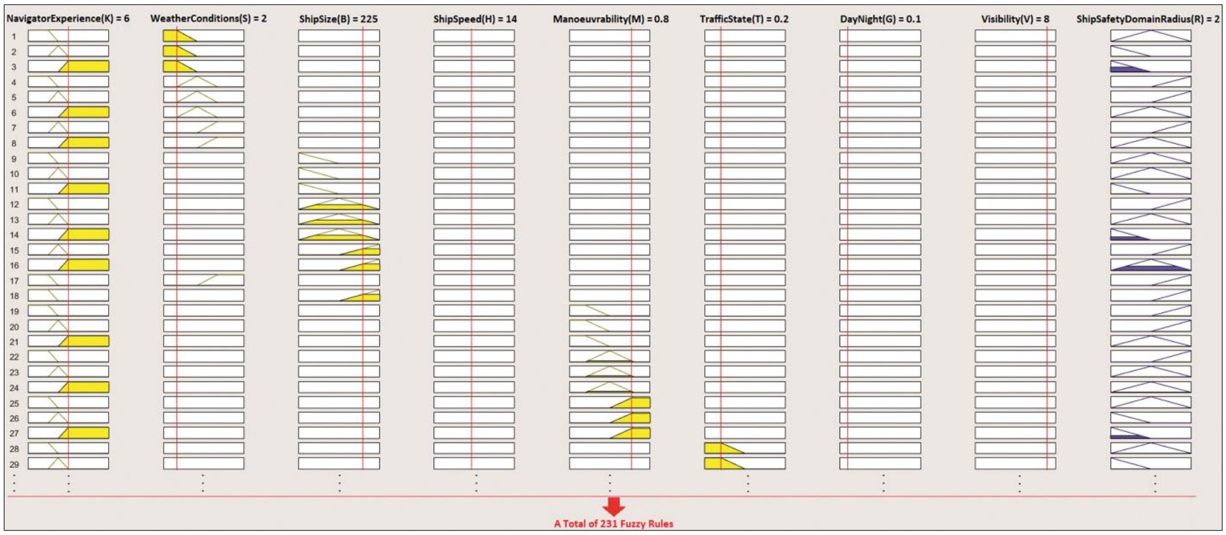


Fig. 12. Virtual environment experiment: fuzzy ship domain calculation.

4.2. Real Environment Experiment: Autonomous Surface Vehicle (ASV) Test

Simulation applications are useful tools for evaluating the performance and accuracy of algorithms, but they cannot replace real-scale tests. Experiments on the virtual environment can include simplification and facilitation, which may prevent evaluating the algorithm performance accurately. For this reason, in addition to the results obtained with simulation, real environment tests are also important in evaluating algorithm performance (Hagen, 2017). Considering this fact, real environment experiments were also conducted in this study. In this section, details of the real environment experiments applied with the ASV (together with NTNU AMOS and MR cooperation) are provided in detail.

In real environment experiments, an ASV named *Telemetron,* owned by MR, was used as the OS and *Munkholmen II*, a tugboat owned by the Trondheim Port Authority, was used as the TS. During the experiments, the ASV connects the On-board System Simulator (OBS) of MR, whose interface is illustrated in Fig. 14, Fig. 15 and Fig. 17, to manage and track the process. AIS-data from both ASV and Munkholmen II is recorded simultaneously for every second by the system. The system also provides the recorded data as a log file, which includes latitude, longitude, COG, SOG, rate of turn, yaw and etc. to allow the analysis of the experiment results. The ASV was utilized by many other studies (Kufoalor et al., 2020; Tengesdal et al., 2020), as well. These studies also demonstrated that the ASV is consistent and reliable for use in experiments.

Experimental tests were carried out on the east side of Munkholmen Island in a 2 Nm diameter area, taking into account the shallow regions and traffic lines, as shown in Fig. 13. This area is located between 63° 26' 28.7" N / 10° 23' 35.8" E, 63° 28' 9.1" N / 10° 23' 41.6" E coordinates at Trondheim Bay (Trondheim Fjord), Norway. It is among the test areas determined by the International Network for Autonomous Ships (INAS) for autonomous ship researches.

The experimental test process applied with ASV consists of two steps, simulation test and field test. Details about these steps are shown in Table 5.

Before the field tests to be applied with the ASV, simulation tests were performed in the OBS developed by MR in order to verify the scenarios by seeing the movements of the ships in advance. The simulation tests revealed that scenarios are applicable for the field tests. Only in the crossing scenario, the OS and TS were seen to be too close to posing a collision risk, so the ship domain and Mx_{toTS} inputs were revised to be applied in the field test. After the simulation test, field tests were applied in a real environment with Telemetron and virtual TS. At the last step, real environmental experiments were completed by applying field tests with Telemetron and Munkholmen II.

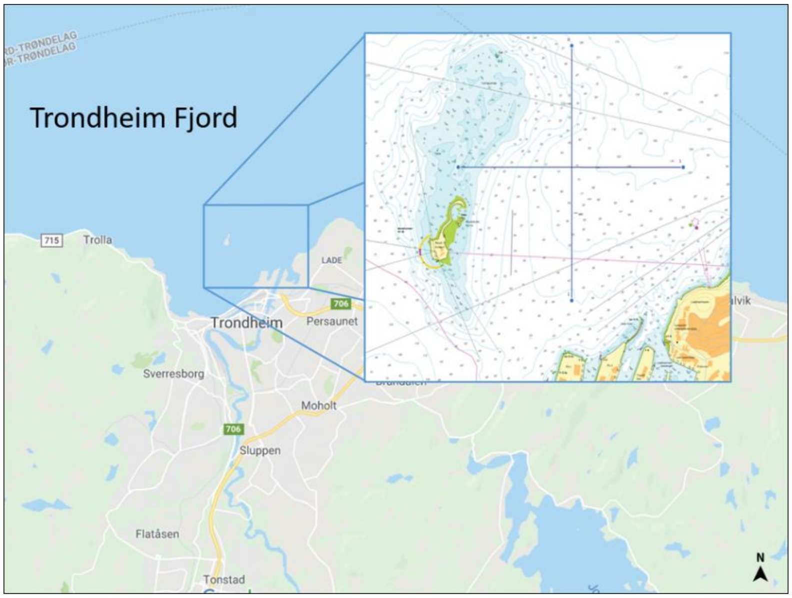


Fig. 13. Real environment experiment: test area.

Table 5
Real environment experiment: steps of the test process.

Day	Location	Step	Implementation	Wind	Vehicles
1. Day	MR office	Simulation test	Bridge simulator test	-	Virtual OS Virtual TS
2. Day	Trondheim Bay	Field test	Field test with virtual TS	1.55 m/s	Telemetron Virtual TS
3. Day	Trondheim Bay	Field test	Field test with real TS	4.18 m/s	Telemetron Munkholmen II

Scenario inputs for this experiment were set as SD_{fuzzy} 0.15 Nm, V_{os} 7 knots, ϕ_{os} 000°, V_{ts} 6 knots, ϕ_{ts} 180°, RB_{ts} 000°, $D_{(os\_ts)}$ 1.5 Nm and Mx_{toTS} 0.5 Nm as shown in Table 6. The movement of the ships leads to the risk of collision ($CPA = 0 < SD_{fuzzy}$). As a result of these inputs, the optimal result provided by the algorithm was as shown in Table 7. According to this output, the ASV made the collision avoidance navigation.

Table 6
Real environment experiment scenario inputs.

	SD [Nm]	V [knot]	$\boldsymbol{\phi}$ [°]	$\boldsymbol{RB_{ts}}$ [°]	$\boldsymbol{D_{(os\_ts)}}$ [Nm]	Mx_{toTS} [Nm]	Initial position
OS	0.15	7	000	-	-	0.5	63° 26' 46.63" N, 010° 24' 3.68" E
TS	-	6	180	000	1.5	-	63° 28' 26.01" N, 010° 24' 3.69" E

Table 7
Real environment experiment: optimal solution produced by the algorithm.

$\boldsymbol{t_{(wp_0\_wp_1)}}$ [m]	$\Delta\Psi_1$ [°]	$\boldsymbol{t_{(wp_1\_wp_2)}}$ [m]	$\Delta\Psi_2$ [°]	$\boldsymbol{t_{(wp_2\_wp_3)}}$ [m]	$\Delta\Psi_3$ [°]	$\boldsymbol{m_{(wp_1\_wp_2)} + m_{(wp_2\_wp_3)}}$ [Nm]
4.8	34	2.4	[-]64	2.68	30	0.59

The trajectory of the ASV to avoid collision in the head-on situation scenario are presented step by step in Fig. 14, Fig. 15, Fig. 17 for simulation test, field test with virtual TS and field test with real TS, respectively. Fig. 16 and Fig. 18 also show the expected and measured course and speed values of the ASV during the collision avoidance navigation applied in the field test with virtual TS and field test with real TS, respectively. The green line in the figures represents the trajectory of the ASV.

As shown in Fig. 14, in the simulation test, ASV performed the collision avoidance navigation as expected in accordance with the output provided by the algorithm. In this scenario, there were no instantaneous changes and deviations in the speed and course of ASV as there was no wind and current effect. This contributed to measure the optimal collision avoidance route length exactly as the algorithm output. Only minor speed changes were observed in the turns, but no effect on the optimal collision avoidance route length.

As shown in Fig. 15, in the field test with virtual TS, the ASV performed the collision avoidance navigation in accordance with the output provided by the algorithm. In this test, due to the effect of wind with 1.55 m/s, instantaneous small deviations were observed in the speed and course of the ASV as shown in Fig. 16. Despite the wind effect, the ASV generally made its navigation in the expected speed and course. The length of the trajectory (0.59 Nm) that occurred at the end of the collision avoidance navigation was exactly the same as the algorithm output.

Fig. 14. Real environment experiment: trajectories of the ships in the simulation test.

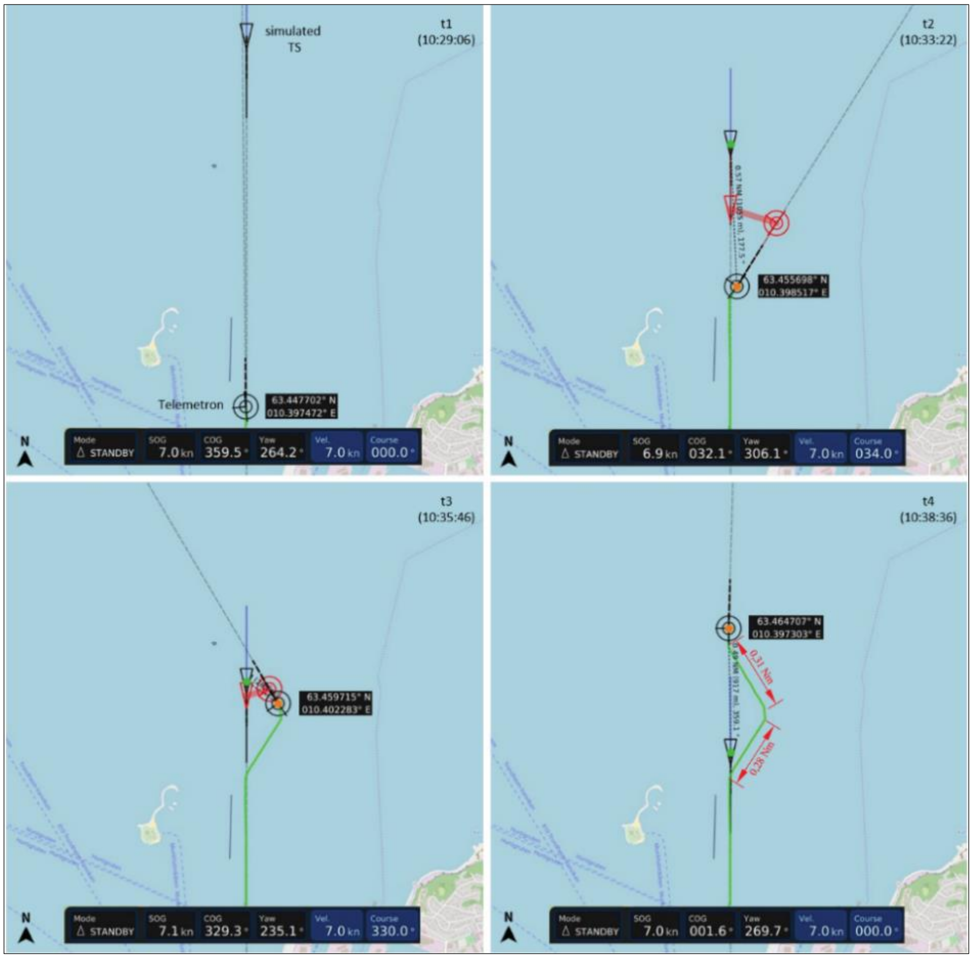


Fig. 15. Real environment experiment: trajectories of the ships in the field test with virtual TS.

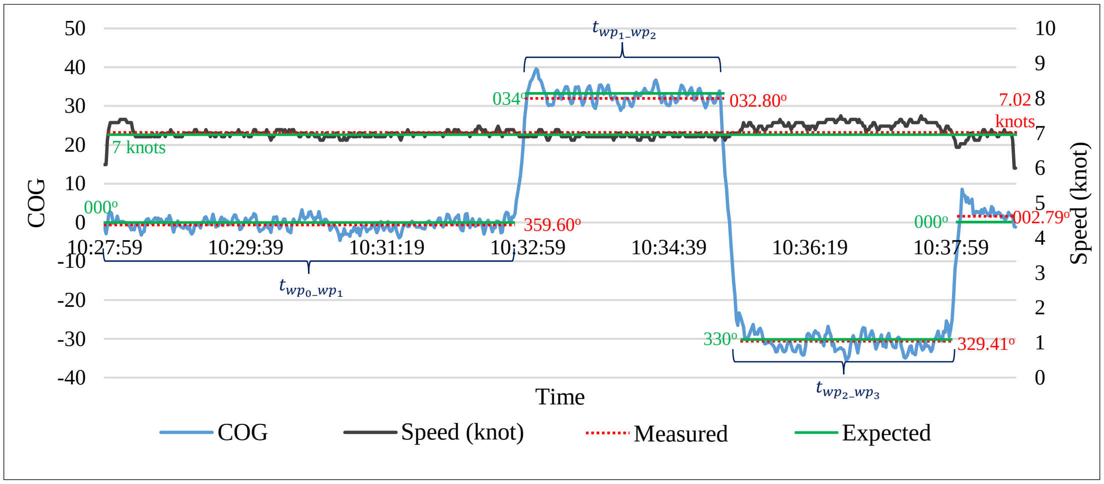


Fig. 16. Real environment experiment: expected and measured course and speed values of the ASV in the field test with virtual TS.

As shown in Fig. 17, in the field test with real TS, the ASV performed the collision avoidance navigation in accordance with the output provided by the algorithm. In this test, due to the effect of wind with 4.18 m/s, instantaneous small deviations were observed in the speed and course of

ASV as shown in Fig. 18. Despite the wind effect, the ASV generally made its navigation safely at the expected speed and course. Although there was a higher wind effect compared to the previous day, the length of the trajectory (0.51 Nm) that occurred at the end of the collision avoidance navigation was measured close to the algorithm output.

5. Discussion

In this part of the study, we aimed to discuss the presented method with the result of the other methods in the related field. The presented method and determined current methods were taken into consideration for comparison. Various encounter types (i.e., head-on and crossing) were established to demonstrate the performance of the presented method. The methods introduced by Tsou et al. (2010), Lazarowska (2014) and Kang et al. (2018), which are AI-based methods, and Lazarowska (2016) and Fiskin et al. (2019), which are deterministic-based methods, were chosen for comparison with different parameter settings, shown in Table 8. The first column in the table provides the case number. The second column shows the encounter type of the cases, followed by the third column and fourth column which present the navigational data inputs belong to the OS and TS, respectively. Table 9 and Table 10, on the other hand, show the numerical results of the comparison scenarios and an in-depth comparison of the methods used for discussion, respectively.

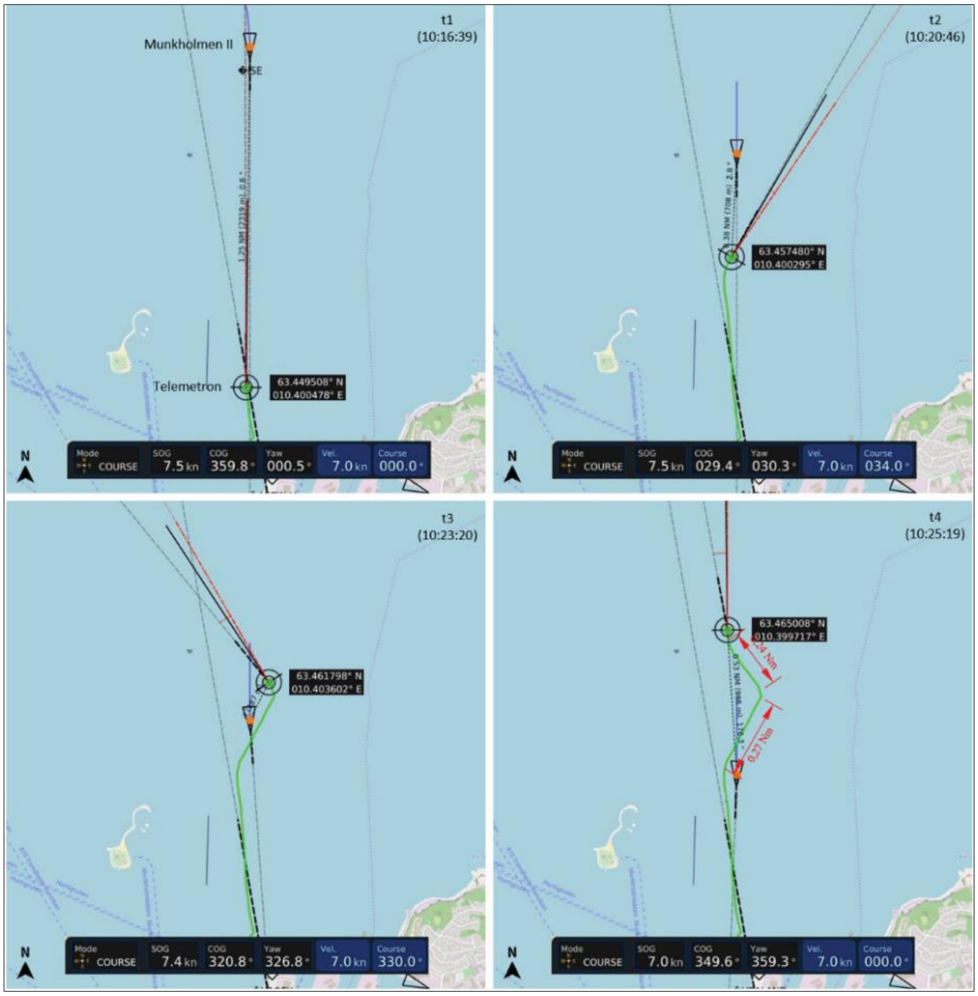


Fig. 17. Real environment experiment: trajectories of the ships in the field test with real TS.

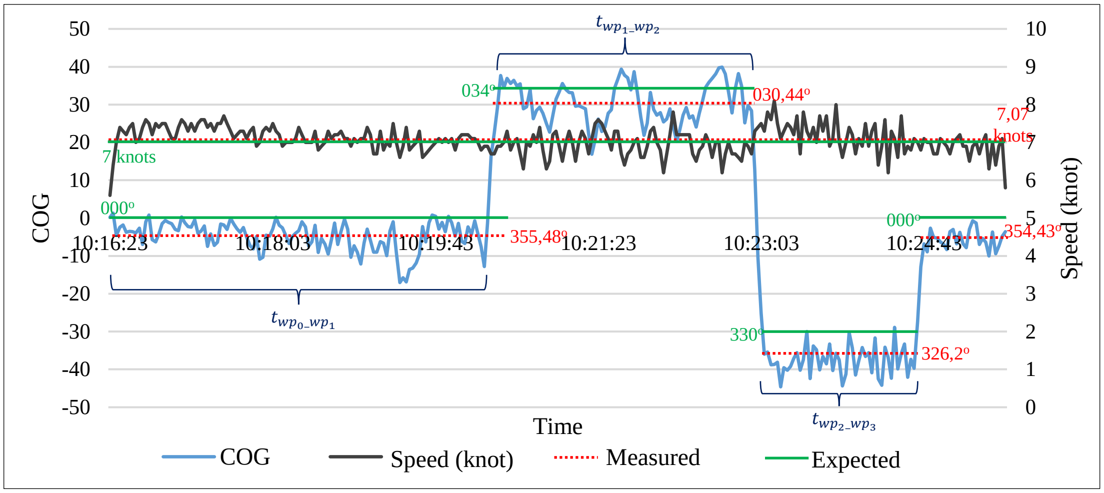


Fig. 18. Real environment experiment: expected and measured course and speed values of the ASV in the field test with real TS.

Table 8

Navigational data of ships for comparison scenarios.

			Navigational data of ships								
			OS				TS				
		Encounter type	ϕ_{os} [°]	V_{os} [kn]	SD [Nm]	RD [Nm]	ϕ_{ts} [°]	V_{ts} [kn]	$D_{(os\text{-}ts)}$ [Nm]	RB_{ts} [°]	Mx_{toTS} [Nm]
Case	1	Head-on	000	30	0.55	-	180	30	10	000	6
	2	Head-on	000	14	1.25	-	180	12	10	000	8
	3	Crossing	000	14	2	-	240	15	32	30	4
	4	Head-on	000	15	2	8.14	180	15	15	000	10

Table 9

Numerical results of comparison scenarios.

		Method	Anti-collision course alteration [°]	Course alteration to return the original route* [°]	Length of the collision avoidance trajectory [Nm]	Computational time [s]
Case	1	ColAv_GA	018	(-)048	3.88	2.63
		PSO	012.40	(-)024.80	6.14	18.05
	2	ColAv_GA	020	(-)050	7.07	5.21
		TBA	014	(-)025	9.22	0.4
		ACO	011	(-)025	9.22	19
	3	ColAv_GA	58	(-)88	4.53	6.8
		GA	46	(-)93	5.55	14-26
	4	ColAv_GA	028	(-)058	9.31	5.51
		WBDA	027.8	(-)056.8	9.39	0.33

Table 10

The comparison of methods used for discussion*.

Reference	(Tsou et al., 2010)	(Lazarowska, 2014)	(Lazarowska, 2016)	(Kang et al., 2018)	(Fiskin et al., 2019)	Method presented here
Method	GA	ACO	TBA	PSO	WBDA	ColAv_GA
Approach type	AI	AI	deterministic	AI	deterministic	AI
Type of manoeuvre	course alteration	course alteration	course alteration	course alteration	course alteration	course alteration / speed change
Number of manoeuvre	single manoeuvre	single manoeuvre	single manoeuvre	single manoeuvre	single manoeuvre	single manoeuvre
Static obstacle	not considered	considered	considered	considered	considered	considered
Dynamic obstacle	considered	considered	considered	considered	considered	considered
Ship domain shape type	circular (around the OS)	hexagon (around the TS)	hexagon (around the TS)	elliptical (around the TS)	circular (around the OS)	circular (around the OS)
Ship domain characteristic	static domain	static domain	static domain	dynamic domain	static domain	static domain
Expression of domain	safety domain	ship domain	ship domain	safety domain	ship domain	ship domain
Safety criterion	domain not be violated	domain not be violated	domain not be violated	domain not be violated	domain not be violated	domain not be violated
Objective function	the length of the trajectory	the length of the trajectory	the length of the trajectory	the length of the trajectory	the length of the trajectory	the length of the trajectory
TS motion	keeps course and speed	keeps course and speed	keeps course and speed	keeps course and speed	keeps course and speed	keeps course and speed
Action range determination to the TS	no	no	no	no	no	yes
Optimization goal option	no	no	no	no	no	yes
Speed change option	no	no	no	no	no	yes
Consulting to experts	no	no	no	no	no	yes
Virtual environment validation test	no	no	no	no	no	yes
Real environment validation test	no	no	no	no	no	yes

*This table was extended from the classifications made by Fiskin et al. (2020, 2019); Lazarowska (2015a).

5.1. Case 1: Comparison with a PSO-Based Method

A comparison of the trajectories of the OSs generated by a particle swarm optimization (PSO)-based method (introduced by Kang et al. (2018)) and ColAv_GA is illustrated in Fig. 19(a). Numerical results produced by both methods are compared in Table 9. It is clear that the solution by the ColAv_GA notably outperformed the PSO-based algorithm. The difference in the length of the trajectories was measured as 2.26 Nm. In terms of the computational time, on the other hand, ColAv_GA reached the solution in a much shorter time. The trajectory produced by the PSO-based method, however, comprised more acute angles. It should be mentioned that this can be an advantage in terms of navigational comfort and not overloading the engine of ships. This situation may not be, however, very important for future concepts (e.g., unmanned ships, autonomous ships).

It should be also stated how we determined the size of the ship domain for this case in order to ensure an accurate comparison. In the PSO-based method, to assess the collision risk, Tam and Bucknall (2010b)'s ship domain concept, which mainly varies according to ship speed, was used. The ship domain determined in the PSO-based method is semi-elliptical and its semi-minor and semi-major axes were assumed to be 0.55 Nm for 30 knots for a head-on encounter situation.

5.2. Case 2: Comparison with an ACO-Based Method and Deterministic Method Called TBA

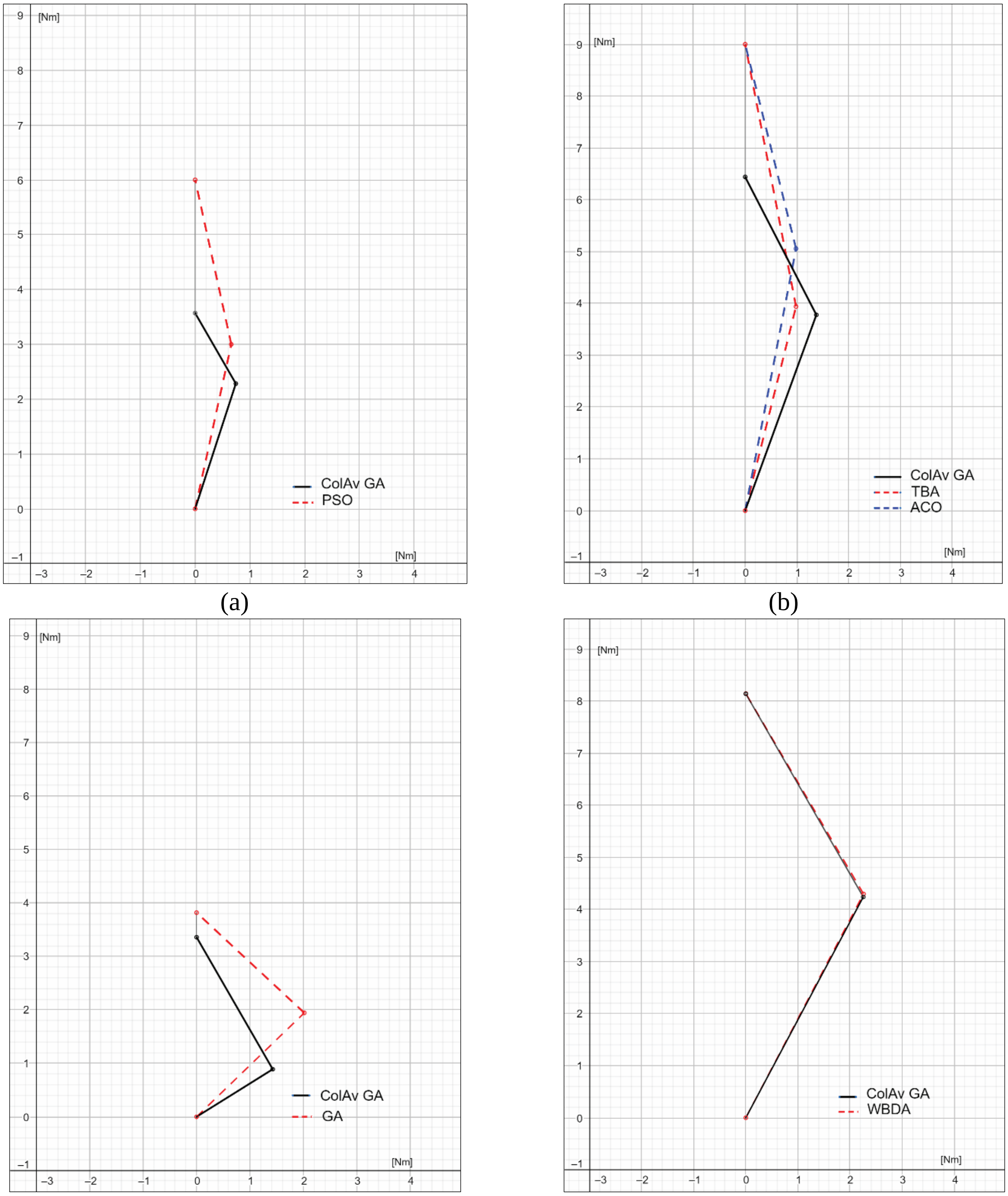
In this case, the results achieved by the ColAv_GA were discussed with the solutions returned by an Ant Colony Optimization (ACO)-based method (proposed by Lazarowska (2014)) and the deterministic method called TBA (introduced by Lazarowska (2016)). The comparison of the trajectories obtained by these three methods for an example of an encounter situation entitled Case 2 in Table 8 is illustrated in Fig.19(b). The collision risk assessment in the ACO-based method and the TBA method was made with a hexagonal ship domain with the longest diagonal line of 1.25 Nm. We determined the size of the ship domain for this exemplary situation considering this value. Fig. 19(b) shows that the trajectory marked by the ColAv_GA is shorter than the trajectories determined by the other two methods. The computational time is in favour of the ColAv_GA compared to the ACO-based method, but we cannot say the same when compared to that of the TBA method. With the advantage of being a deterministic method, TBA returned the solution in a much shorter time, as shown in Table 9.

5.3. Case 3: Comparison with a GA-Based Method

Case 3 provides a comparison between the solutions determined by the ColAv_GA and a GA-based method (proposed by Tsou et al. (2010)). Numerical results produced by both methods are compared in Table 9. It is obvious that the result by the ColAv_GA outperformed the GA-based method which is illustrated in Fig. 19(c). The difference of the length of the trajectories was calculated as 1.02 Nm. In terms of the computational time, on the other hand, ColAv_GA reached the solution in a much shorter time.

5.4. Case 4: Comparison with a Deterministic Method Called WBDA

A comparison of the trajectories determined by the ColAv_GA and Web-Based Deterministic Algorithm (WBDA) (developed by (Fiskin et al. (2019)) is shown in Fig. 19(d). The trajectory provided by ColAv_GA consists of two-course alteration, the first one of 028 degrees and the second one of (-)058 degrees, while the solution obtained by the WBDA also comprises of two-course alteration with values of 027.8 degrees and (-)056.8 degrees, respectively. The trajectories provided by both algorithms are almost identical with a minor difference of 0.08 Nm in favour of the ColAv_GA. In terms of the computational time, however, with the advantage of being a deterministic method, WBDA reached the solution in a much shorter time, as demonstrated in Table 9. In this exemplary case, we should provide the return distance (RD) which is the final point distance to return the OS to its original route. In the WBDA method, the RD is given for the OS. The WBDA method was also developed by some of the authors in this study. Firstly, this case was executed in the ColAv_GA, then we calculated the RD distance using the sine theorem and used it as input for WBDA to ensure an accurate comparison.



(a) (b)

(c) (d)

Fig. 19. OS trajectories obtained by the methods in comparison study.

To sum up, the ColAv_GA outperformed in terms of computational time compared to AI-based methods. But we cannot say the same against deterministic methods. The comparison study demonstrated that deterministic methods return solutions in a much shorter time. However, it should be emphasized that the computational time of ColAv_GA, which is about 2-7 seconds, is also at an acceptable level for real-time applications.

6. Conclusion

In this study, in encounter situations at sea that is considered to be a risk of collision, collision avoidance route planning and optimization was performed for give-way vessels. For this purpose, GA, a meta-heuristic algorithm, and fuzzy logic-based solution algorithm, called ColAv_GA, was developed.

In conclusion, the benefits and contributions of this research can be listed as follows:

- The algorithm meets COLREG requirements.
- The algorithm produces identical solutions in every run.
- Execution time of the algorithm is very short. This allows it to be used in real-time applications.
- Variable constraints and scenario inputs were determined as a result of interviews with experts.
- Algorithm was verified in a real environment with ASV, and in a virtual environment with a bridge simulator.
- Fuzzy logic approach was used to calculate the ship domain. Thus, determining the size of the ship domain is removed from subjectivity by linking to rules and objective principles.
- Taking into account the COLREG rule 16, determining the point where the collision avoidance manoeuvre will be applied determined by the system user considering the distance of the TS. Thus, the OS is enabled to manoeuvre at a safe distance to the TS.
- Considering the COLREG rule 8(d), manoeuvring to return to the original route is made after the TS is considered as clear.

- The algorithm can produce solutions according to the navigator's optimization goal (shortest path or MxtoTS).
- The algorithm has the capability to produce optimal results with course alteration and speed change manoeuvre options.

There are some limitations to the implementation of the research. In this research, the interview method was used as a data collection method in the qualitative research process. Since it is not possible to reach the whole target population in the interview method, the research is carried out through the sample population. The sample size consists of a total of 10 people including 1 pilot interview. In this study, the sample size considered sufficient for interviews that key data are intended to be obtained was reached. However, the current sample size can be considered as one of the limitations of qualitative research. In the interview method, certain answers cannot be obtained from some participants in line with the objectives of the research. This situation emerges as another limitation of the research. Another limitation is the motion of the TS. Changes in motion strategy of the TS are not taken into account. If the course or speed change of the TS is detected, a reactive approach with recalculation is made according to new data.

In order to further develop the system, weights of fuzzy rules can be defined according to the order of importance of ship domain input variables utilizing multi-criteria decision-making methods (AHP, ANP, etc.) for further studies. Moreover, for future studies, a comparative study can be conducted to examine the effects of different ship domain types on the collision avoidance route.

Acknowledgments

The authors are grateful to Maritime Robotics AS for involving them in the real environment experiments leading to substantial results. We would like to thank Stephanie Kemna, Arild Hepso and Caio Swan de Freitas from Maritime Robotics and Trondheim Port Authority for their efforts to make the sea trials with the Telemetron possible. And special thanks to D.Kwame Minde Kufoalor from NTNU for his useful comments and efforts on this study.

This study was granted by, the Scientific Research Coordination Unit of Dokuz Eylül University (Project Number: 2017.KB.SOS.005), the Scientific and Technological Research Council of Turkey (TUBITAK) (Project Number: 1059B141800295), and the Research Council of Norway through AMOS grant 223254.

Appendix A. Other situation results and pseudo code of the algorithm

Table A.1

Other encounter types results of both virtual and real environment experiments.

Virtual environment experiment: Other encounter types results

Inputs

		SD [Nm]	V [knot]	ϕ [°]	RB_{ts} [°]	$D_{(os\_ts)}$ [Nm]	Mx_{toTS} [Nm]
Crossing	OS	2	14	000	-	-	7
	TS	-	15	240	030	20	-
Overtaking	OS	2	16	000	-	-	5
	TS	-	7	000	000	12	-

Outputs

	$t_{(wp_0\_wp_1)}$ [m]	$\Delta\Psi_1$ [°]	$t_{(wp_1\_wp_2)}$ [m]	$\Delta\Psi_2$ [°]	$t_{(wp_2\_wp_3)}$ [m]	$\Delta\Psi_3$ [°]	$m_{(wp_1\_wp_2)} + m_{(wp_2\_wp_3)}$ [Nm]
Crossing	31.2	54	10.2	[-]84	17.47	30	6.6
Overtaking	46.8	14	38.4	[-]44	18.58	30	15.19

Real environment experiment: Other encounter types results

Inputs

		SD [Nm]	V [knot]	ϕ [°]	RB_{ts} [°]	$D_{(os\_ts)}$ [Nm]	Mx_{toTS} [Nm]	Initial position
Crossing	OS	0.2	7	090		-	0.5	63° 27' 30" N, 010° 22' 10.2" E
	TS	-	6	000	040	1	-	63° 26' 46.8" N, 010° 24' 3.69" E
Overtaking	OS	0.2	9	000		-	0.3	63° 26' 46.8" N, 010° 24' 3.69" E
	TS	-	4	000	000	0.5	-	63° 27' 16.2" N, 010° 24' 3.69" E

Outputs

	$t_{(wp_0\_wp_1)}$ [m]	$\Delta\Psi_1$ [°]	$t_{(wp_1\_wp_2)}$ [m]	$\Delta\Psi_2$ [°]	$t_{(wp_2\_wp_3)}$ [m]	$\Delta\Psi_3$ [°]	$m_{(wp_1\_wp_2)} + m_{(wp_2\_wp_3)}$ [Nm]
Crossing	3.6	58	2.4	[-]88	4.07	30	0.75
Overtaking	3	32	3.6	[-]54	5.09	22	1.3

Table A.2

Pseudo code of the algorithm.

```
function GA
1   begin
2   // function create_initial_population
3   {
4       set population size
5       for each individual
6       {
7               assign gene randomly
8               check for any overlap
9               if (yes)
10              {
11                  back to step 7
12              }
13              else
14              {
15                  assign gene
16              }
17      }
18      return population
19  }
20  // function calculate_fitness
21  {
22      get population list
23      for each individual
24      {
25              calculate $f(x) = min \sum_{i=1}^{n-1} m_i$
26      }
27      return fitness value
28  }
29  best solution$_o$ = min(calculate_fitness$_o$)
30  determine the best individual (elitism)
31  iteration=0
32  repeat
33  {
34      // function roulette_wheel_selection
35      {
36              get population list
37              for each fitness calculated individual
38              {
39                  determine selection probabilities ($P_i = f(x_i)/\sum_{i=1}^{N_P} f(x_i)$)
40              }
41              spin the wheel and select individuals
42      }       return population
43      // function crossover
44      {
45              get the list of individuals in the match pool
46              match individuals randomly
47              determine crossover rate
48              for each parent
49              if (rand(0,1) < C_R)
50              {
51                  determine crossover point randomly
52                  change genes between pairs at crossover point
53              }
54              return child individuals
55      }
56      // fonksiyon mutation
57      {
58              get the list of child individuals
59              determine mutation rate
60              for each individual
61              if (rand(0,1) < M_R)
62              {
63                  specify the gene to be mutated randomly and reverse it
64              }
65              return new population
66      }
67  }
68  calculate fitness
69  best solution$_t$ = min(calculate_fitness$_t$)
70  if (best solution$_t$ < best solution$_o$)
71  {
72      best solution$_o$ = best solution$_t$
73  }
74  iteration ++
75  if (iteration > iteration number)
76  {
77      end
78  }
79  else
80  {
81      back to step 32
82  }
83  end
```

References

Akkoyunlu, M.C., Engin, O., 2011. Optimization problems solving with using of discrete harmony search algorithm: a review. J. Fac.Eng.Arch. Selcuk Univ 2626, 1300–5200.

Blum, C., Roli, A., 2008. Hybrid Metaheuristics: An Introduction., in: Hybrid Metaheuristics. Springer, Germany, pp. 1–30.

Bowo, L.P., Furusho, M., 2019. Usability of human error assessment and reduction technique with a 4M framework (HEART–4M) – A case study on ship grounding accidents. J. ETA Marit. Sci. 7, 266–279. https://doi.org/10.5505/jems.2019.54775

Brcko, T., Švetak, J., 2013. Fuzzy reasoning as a base for collision avoidance decision support system. PROMET - Traffic&Transportation 25, 555–564. https://doi.org/10.7307/ptt.v25i6.1183

Büyükyazıcı, M., Taşar, E., 2011. Optimal retention limit with Monte Carlo stochastic optimization. J. Stat. Stat. Actuar. Sci. 4, 1–8.

Candeloro, M., Lekkas, A.M., Sørensen, A.J., 2017. A Voronoi-diagram-based dynamic path-planning system for underactuated marine vessels. Control Eng. Pract. 61, 41–54. https://doi.org/10.1016/j.conengprac.2017.01.007

Cheng, X., Liu, Z., 2007. Trajectory optimization for ship navigation safety using genetic annealing algorithm. Proc. - Third Int. Conf. Nat. Comput. ICNC 2007 4, 385–389. https://doi.org/10.1109/ICNC.2007.783

Cheng, X.D., Liu, Z.Y., Zhang, X.T., 2007. Trajectory optimization for ship collision avoidance system using genetic algorithm, in: OCEANS 2006 - Asia Pacific. pp. 2–6.

Coldwell, T.G., 1983. Marine traffic behaviour in restricted waters. J. Navig. 36, 430–444. https://doi.org/10.1017/S0373463300039783

Davis, P. V., Dove, M.J., Stockel, C.T., 1980. A computer simulation of marine traffic using domains and arenas. J. Navig. 33, 215–222. https://doi.org/10.1017/S0373463300035220

Dinh, G.H., Im, N., 2016. The combination of analytical and statistical method to define polygonal ship domain and reflect human experiences in estimating dangerous area. Int. J. e-Navigation Marit. Econ. 4, 97–108. https://doi.org/10.1016/j.enavi.2016.06.009

Esmer, S., 2009. Optimization of logistics processes at the container terminals and a simulation model. Dokuz Eylül University.

Fiskin, R., 2019. Route planning and optimization for maritime collision avoidance. Dokuz Eylül University.

Fiskin, R., Kisi, H., Nasibov, E., 2018. A research on techniques, models and methods proposed for ship collision avoidance path planning problem. Trans. R. Inst. Nav. Archit. Part A Int. J. Marit. Eng. 160, 187–205. https://doi.org/10.3940/rina.ijme.2018.a2.476

Fiskin, R., Nasibov, E., Yardimci, M.O., 2020. A knowledge-based framework for two-dimensional (2D) asymmetrical polygonal ship domain. Ocean Eng. 202, 107187. https://doi.org/10.1016/j.oceaneng.2020.107187

Fiskin, R., Nasibov, E., Yardimci, M.O., 2019. Deterministic-based ship anti-collision route optimization with web-based application. Trans. R. Inst. Nav. Archit. Part A Int. J. Marit. Eng. 161, 345–356. https://doi.org/10.3940/rina.ijme.2019.a4.537

Goodwin, E.M., 1975. A statistical study of ship domains. J. Navig. 28, 328–344. https://doi.org/10.1017/S0373463300041230

Grinyak, V.M., Devyatisil'nyi, A.S., 2016. Fuzzy collision avoidance system for ships. J. Comput. Syst. Sci. Int. 55, 249–259. https://doi.org/10.1134/S106423071601007X

Hagen, I.B., 2017. Collision avoidance for ASVs using model predictive control. Norwegian University of Science and Technology.

Hao, J.L., Zhao, L.N., Hu, J.F., Yang, X.B., 2007. Decision-making model for multi-ship collision avoidance based on adaptive genetic algorithm, in: International Conference on

Transportation Engineering 2007. pp. 2–4.

Hongdan, L., Sheng, L., Lanyong, Z., 2015a. Ship collision avoidance path planning strategy based on quantum bacterial foraging algorithm, in: International Conference on Electrical, Computer Engineering and Electronics. pp. 612–621. https://doi.org/10.2991/icecee-15.2015.124

Hongdan, L., Sheng, L., Lanyong, Z., Zhenguo, G., 2014. The application research with particle swarm bacterial foraging intelligent algorithm in ship collision avoidance, in: International Conference on Mechatronics, Control and Electronic Engineering. pp. 69–74. https://doi.org/10.2991/mce-14.2014.15

Hongdan, L., Sheng, L., Zhuo, Y., 2015b. Application of adaptive wolf colony search algorithm in ship collision avoidance. Int. J. Simul. Syst. Sci. Technol. 16, 14.1-14.7. https://doi.org/10.5013/IJSSST.a.16.2A.14

Jingsong, Z., Zhaolin, W., Fengchen, W., 1993. Comments on ship domains. J. Navig. 46, 422–436. https://doi.org/10.1017/S0373463300011875

Johansen, T.A., Perez, T., Cristofaro, A., 2016. Ship collision avoidance and COLREGS compliance using simulation-based control behavior selection with predictive hazard assessment. IEEE Trans. Intell. Transp. Syst. 17, 3407–3422. https://doi.org/10.1109/TITS.2016.2551780

Kang, Y.T., Chen, W.J., Zhu, D.Q., Wang, J.H., Xie, Q.M., 2018. Collision avoidance path planning for ships by particle swarm optimization. J. Mar. Sci. Technol. 26, 777–786. https://doi.org/10.6119/JMST.201812_26(6).0003

Kaveh, A., Ilchi Ghazaan, M., 2018. Meta-heuristic algorithms for optimal design of real-size structures. Springer, Switzerland.

Kelton, W.D., Sadowski, R.P., Sadowski, D.A., 2003. Simulation with Arena. McGraw-Hill, USA.

Kufoalor, D.K.M., Johansen, T.A., Brekke, E.F., Hepsø, A., Trnka, K., 2020. Autonomous maritime collision avoidance: Field verification of autonomous surface vehicle behavior in challenging scenarios. J. F. Robot. 37, 387–403. https://doi.org/10.1002/rob.21919

Lazarowska, A., 2016. A Trajectory Base Method for Ship's Safe Path Planning. Procedia Comput. Sci. 96, 1022–1031. https://doi.org/10.1016/j.procs.2016.08.118

Lazarowska, A., 2015a. Ship's trajectory planning for collision avoidance at sea based on ant colony optimisation. J. Navig. 68, 291–307. https://doi.org/10.1017/S0373463314000708

Lazarowska, A., 2015b. Parameters influence on the performance of an ant algorithm for safe ship trajectory planning, in: IEEE 2nd International Conference on Cybernetics, CYBCONF 2015. pp. 140–145. https://doi.org/10.1109/CYBConf.2015.7175921

Lazarowska, A., 2014. Ant Colony Optimization based navigational decision support system. Procedia Comput. Sci. 35, 1013–1022. https://doi.org/10.1016/j.procs.2014.08.187

Lazarowska, A., 2012. Decision support system for collision avoidance at sea. Polish Marit. Res. 19, 19–24. https://doi.org/10.2478/v10012-012-0018-2

Lee, H., Choi, S., Jung, H., Park, B.B., Son, S.H., 2019. A route guidance system considering travel time unreliability. J. Intell. Transp. Syst. Technol. Planning, Oper. 23, 282–299. https://doi.org/10.1080/15472450.2018.1542303

Lee, Y. Il, Kim, S.G., Kim, Y.G., 2015. Fuzzy relational product for collision avoidance of autonomous ships. Intell. Autom. Soft Comput. 21, 21–38. https://doi.org/10.1080/10798587.2014.914273

Li, S., Liu, J., Negenborn, R.R., 2019a. Distributed coordination for collision avoidance of multiple ships considering ship maneuverability. Ocean Eng. 181, 212–226. https://doi.org/10.1016/j.oceaneng.2019.03.054

Li, S., Liu, J., Negenborn, R.R., Ma, F., 2019b. Optimizing the joint collision avoidance operations of multiple ships from an overall perspective. Ocean Eng. 191, 106511. https://doi.org/10.1016/j.oceaneng.2019.106511

Li, W., Ma, W., 2016. Simulation on vessel intelligent collision avoidance based on artificial fish swarm algorithm. Polish Marit. Res. 23, 138–143. https://doi.org/10.1515/pomr-2016-0058

Li, Y., Li, K., Tong, S., 2019. Finite-time adaptive fuzzy output feedback dynamic surface control for MIMO nonstrict feedback systems. IEEE Trans. Fuzzy Syst. 27, 96–110. https://doi.org/10.1109/TFUZZ.2018.2868898

Lisowski, J., 2000. Multistage ship's optimal control in collision situations using a neural network, in: 2nd International Conference on Safe Navigation Beyond 2000. pp. 1–10.

Lisowski, J., Rak, A., Czechowicz, W., 2000. Neural network classifier for ship domain assessment. Math. Comput. Simul. 51, 399–406. https://doi.org/10.1016/s0378-4754(99)00132-9

Liu, Y., Yang, C., Du, X., 2007. A multiagent-based simulation system for ship collision avoidance. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 4681 LNCS, 316–326. https://doi.org/10.1007/978-3-540-74171-8_31

Mostefa, M.S., 2014. The branch-and-bound method, genetic algorithm, and dynamic programming to determine a safe ship trajectory in fuzzy environment. Procedia Comput. Sci. 35, 348–357. https://doi.org/10.1016/j.procs.2014.08.115

Naeem, W., Irwin, G.W., Yang, A., 2012. COLREGs-based collision avoidance strategies for unmanned surface vehicles. Mechatronics 22, 669–678. https://doi.org/10.1016/j.mechatronics.2011.09.012

Perera, L.P., Carvalho, J.P., Guedes Soares, C., 2011. Fuzzy logic based decision making system for collision avoidance of ocean navigation under critical collision conditions. J. Mar. Sci. Technol. 16, 84–99. https://doi.org/10.1007/s00773-010-0106-x

Perera, L.P., Carvalho, J.P., Soares, C.G., 2010. Fuzzy-logic based parallel collisions avoidance decision formulation for an ocean navigational system, IFAC Proceedings Volumes (IFAC-PapersOnline). IFAC. https://doi.org/10.3182/20100915-3-DE-3008.00044

Perera, L.P., Ferrari, V., Santos, F.P., Hinostroza, M.A., Guedes Soares, C., 2015. Experimental evaluations on ship autonomous navigation and collision avoidance by intelligent guidance. IEEE J. Ocean. Eng. 40, 374–387. https://doi.org/10.1109/JOE.2014.2304793

Pietrzykowski, Z., 2008. Ship's fuzzy domain - A criterion for navigational safety in narrow fairways. J. Navig. 61, 499–514. https://doi.org/10.1017/S0373463308004682

Simsir, U., Amasyali, M.F., Bal, M., Çelebi, U.B., Ertugrul, S., 2014. Decision support system for collision avoidance of vessels. Appl. Soft Comput. J. 25, 369–378. https://doi.org/10.1016/j.asoc.2014.08.067

Smierzchalski, R., Michalewicz, Z., 1998. Adaptive modeling of a ship trajectory in collision situations at sea. Proc. IEEE Conf. Evol. Comput. ICEC 342–347. https://doi.org/10.1109/icec.1998.699756

Su, C.M., Chang, K.Y., Cheng, C.Y., 2012. Fuzzy decision on optimal collision avoidance measures for ships in vessel traffic service. J. Mar. Sci. Technol. 20, 38–48.

Szlapczynski, R., 2015. Evolutionary planning of safe ship tracks in restricted visibility. J. Navig. 68, 39–51. https://doi.org/10.1017/S0373463314000587

Szlapczynski, R., 2008. A new method of planning collision avoidance manoeuvres for multi-target encounter situations. J. Navig. 61, 307–321. https://doi.org/10.1017/S0373463307004638

Szlapczyński, R., 2007. Determining the optimal course alteration manoeuvre in a multi-target encounter situation for a given ship domain model. Annu. Navig. 12, 75–85.

Szlapczynski, R., Szlapczynska, J., 2017. Review of ship safety domains: Models and applications. Ocean Eng. 145, 277–289. https://doi.org/10.1016/j.oceaneng.2017.09.020

Tam, C., Bucknall, R., 2010a. Path-planning algorithm for ships in close-range encounters. J. Mar. Sci. Technol. 15, 395–407. https://doi.org/10.1007/s00773-010-0094-x

Tam, C., Bucknall, R., 2010b. Collision risk assessment for ships. J. Mar. Sci. Technol. 15, 257–270. https://doi.org/10.1007/s00773-010-0089-7

Tengesdal, T., Johansen, T.A., Brekke, E., 2020. Risk-based Autonomous maritime collision avoidance considering obstacle intentions, in: FUSION 2020.

Tsou, M.C., 2019. Big data analytics of safety assessment for a port of entry: A case study in Keelung Harbor. Proc. Inst. Mech. Eng. Part M J. Eng. Marit. Environ. 233, 1260–1275. https://doi.org/10.1177/1475090218805245

Tsou, M.C., Hsueh, K.C., 2010. The study of ship collision avoidance route planning by ant colony algorithm. J. Mar. Sci. Technol. 18, 746–756.

Tsou, M.C., Kao, S.L., Su, C.M., 2010. Decision support from genetic algorithms for ship collision avoidance route planning and alerts. J. Navig. 63, 167–182. https://doi.org/10.1017/S037346330999021X

Wei, Z., Zhou, K., Wei, M., 2015. Decision-making in ship collision avoidance based on cat-swarm biological algorithm, in: International Conference on Computational Science and Engineering. pp. 114–122. https://doi.org/10.2991/iccse-15.2015.20

Xu, Q., 2014. Collision avoidance strategy optimization based on danger immune algorithm. Comput. Ind. Eng. 76, 268–279. https://doi.org/10.1016/j.cie.2014.08.010

Xu, Q., Wang, N., 2014. A survey on ship collision risk evaluation. Promet - Traffic - Traffico 26, 475–486. https://doi.org/10.7307/ptt.v26i6.1386

Xu, Q., Zhang, C., Wang, N., 2014. Multiobjective optimization based vessel collision avoidance strategy optimization. Math. Probl. Eng. https://doi.org/10.1155/2014/914689

Yang, L.J., Hong, B.G., Inoue, K., Sadakane, H., 2010. Experimental study on braking force characteristics of tugboat. J. Hydrodyn. 22, 343–348. https://doi.org/10.1016/S1001-6058(09)60216-X

Yardimci, M.O., Fiskin, R., Nasibov, E., 2019. A fuzzy rule-based approach to determine an asymmetrical polygonal ship domain, in: Proceedings - 2019 Innovations in Intelligent Systems and Applications Conference, ASYU 2019. pp. 0–3. https://doi.org/10.1109/ASYU48272.2019.8946447

Yavin, Y., Frangos, C., Miloh, T., Zilman, G., 1998. Collision avoidance by a ship with a moving obstacle: Computation of feasible command strategies. J. Optim. Theory Appl. 98, 243–244. https://doi.org/10.1023/A:1022653317941

Zhang, L., Meng, Q., 2020. Response to the discussion by Montewka Jaku, Gil Mateusz and Wróbel Krzysztof on the article by Zhang & Meng entitled “Probabilistic ship domain with applications to ship collision risk assessment [Ocean Eng. 186 (2019) 106130]. Ocean Eng. 209, 107471. https://doi.org/10.1016/j.oceaneng.2020.107471

Zhu, X., Xu, H., Lin, J., 2001. Domain and its model based on neural networks. J. Navig. 54, 97–103. https://doi.org/10.1017/S0373463300001247