A ship collision avoidance system for human-machine cooperation during collision avoidance

Huang, Yamin; Chen, Linying; Negenborn, Rudy R.; van Gelder, P. H.A.J.M.

DOI
10.1016/j.oceaneng.2020.107913

Publication date
2020

Document Version
Final published version

Published in
Ocean Engineering

Citation (APA)
Huang, Y., Chen, L., Negenborn, R. R., & van Gelder, P. H. A. J. M. (2020). A ship collision avoidance system for human-machine cooperation during collision avoidance. *Ocean Engineering*, *217*, Article 107913. https://doi.org/10.1016/j.oceaneng.2020.107913

Ocean Engineering 217 (2020) 107913

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng

A ship collision avoidance system for human-machine cooperation during collision avoidance

Yamin Huang [a,b,d], Linying Chen [c,e,*], Rudy R. Negenborn [e], P.H.A.J.M. van Gelder [d]

[a] Intelligent Transport System Research Center, Wuhan University of Technology, Wuhan, PR China
[b] National Engineering Research Center for Water Transport Safety, Wuhan, PR China
[c] School of Navigation, Wuhan University of Technology, Wuhan, PR China
[d] Safety and Security Science Group, Faculty of Technology, Policy and Management, Delft University of Technology, Delft, the Netherlands
[e] Department of Maritime and Transport Technology, Faculty of Mechanical, Maritime and Materials Engineering, Delft University of Technology, Delft, the Netherlands

ARTICLE INFO

Keywords:
Human-machine interaction
Collision avoidance system
Under-actuated ship
Shared decision-making
MASS

ABSTRACT

Maritime Autonomous Surface Ships (MASS) attract increasing attention in recent years. Researchers aim at developing fully autonomous systems that replace the role of human operators. Studies either focus on supporting conflict/collision detection (for manned ships) or solving conflict automatically (for unmanned ships). The cooperation between human and machine has been less focused on in existing studies. However, this type of cooperation is essential both in practice and in the future: firstly, demands on navigational assistance are still strong for supporting navigators in manned ships; secondly, MASS with different autonomy levels require increasing cooperation between human operators and machines, e.g. monitoring automation, remotely controlling the ship, etc.; thirdly, the intelligence of human and the machines is highly complementary. Moreover, fully autonomous ships cannot replace all the manual ships overnight. Therefore, the future waterborne transport system will be a system in which both human-operated vessels and autonomous vessels exist.

In this article, we firstly provide an overview of existing modes of human-machine interaction (HMI) during ship collision avoidance. Then, we propose a framework of HMI oriented Collision Avoidance System (HMI-CAS) whose decision-making process is interpretable and interactive for human operators. The HMI-CAS facilitates automatic collision avoidance and enables the human operators to take over the control of the MASS safely. Moreover, the proposed framework acknowledges the under-actuated feature of ships. Simulations are carried out to demonstrate the proposed HMI-CAS. The results show that the proposed HMI-CAS can not only control the under-actuated MASS to avoid collision automatically but also share the decision-making with human operators and support the operators to control the MASS.

1. Introduction

1.1. Background

Maritime Autonomous Surface Ships (MASS) have been receiving numerous attention and discussions from the public and the academic, which is expected to reduce human errors on-board, to improve navigational safety, and to increase the efficiency of waterborne traffic systems. Hence, various prototypes of MASS have been proposed in recent years (Liu et al., 2016; Schiaretti et al., 2017). The ultimate aim of relevant studies is at removing human operators in the control loop, which is, however, still unrealistic. Teaching the automation to understand regulations is still challenging, and the trust of the public on the autonomous ships is still questionable.

In fact, the machines and the human are needless to be adversarial. Human intelligence and machine intelligence are complementary, where the first one is good at interpreting navigational regulations, and the second one has advantages of computing powers (HHuang et al., 2020a). Communication and cooperation between human and automation become important in the development of autonomous vehicles (Koopman and Wagner, 2017). In other words, the interactions between the human and the machine in collision avoidance are essential for developing relevant automatic (or even autonomous) systems in various MASS, which were rarely discussed in existing research.

* Corresponding author. School of Navigation, Wuhan University of Technology, Wuhan, PR China.
E-mail addresses: yaminhuang@whut.edu.cn, y.m.huang@outlook.com (Y. Huang), linyingchen@whut.edu.cn, chenlinying@outlook.com (L. Chen).

https://doi.org/10.1016/j.oceaneng.2020.107913
Received 22 January 2020; Received in revised form 30 June 2020; Accepted 6 August 2020
Available online 18 September 2020

1.2. Motivation

Fig. 1 demonstrates the system that integrates two loops without considering Human-Machine Interactions (HMIs): one is with humans on board (noted as the manual loop); the other is without humans (noted as the unmanned loop). A switch is employed to change the control mode between two loops.

In the unmanned loop, the guidance system is in charge of finding a collision-free solution, and the control system is designed to execute this solution (e.g. desired forces, trajectories, etc.) Since humans on board are not considered in this loop, the outputs of the guidance system or the control system are usually difficult for humans to read, to intervene, and to implement. For instance, if the guidance system outputs the desired forces in each direction, how to allocate these forces to steers is a question for human operators; alternatively, the system might offer a collision-free trajectory or a set of steers that are readable for human, but steering the ship to each waypoint or following each planned steer sticking to a planned schedule is also challenging for human operators.

If a human operator wants to intervene in the automatic collision avoidance process, e.g. the automation will violate regulations, etc., the operator needs to switch to the manual loop. Then, the operator has to analyze the encounters again, find out possible collision-free solutions, and implement the solution by himself/herself. In this hybrid system, the interaction between humans and machines is little, and there is a lack of information exchange between them.

Neither of the manned system nor the unmanned system in Fig. 1 is perfect. If the existing manned system is reliable, there would not be room for developing unmanned systems. In many accidents, the operators take evasive actions too late or no actions in urgent cases (Graziano et al., 2016). Besides, the traditional navigation assistant systems lack the function of suggesting collision-free solutions (Tam et al., 2009), which is a strength of the unmanned system. On the other hand, the unmanned loop is also not perfect, in particular, how can the unmanned systems understand regulations is still an open question (Campbell et al., 2012), where human knowledge is indispensable.

Therefore, a connection between these two loops could be a "win-win" solution, where a system incorporating HMIs is needed. However, to the best of our knowledge, studies on the manned ship or the unmanned ship do not incorporate the cooperation between human and machines in finding a collision-free solution. In manned-ship studies, the human is supported by the machines to detect dangers (Goerlandt et al., 2015), while finding a collision-free solution is in charge of human operators. In unmanned-ship studies, the machine is supposed to be highly intelligent, and there is no interface for operators to interact/cooperate with the machine (Zheng, 2016). That is, the human operator or the machine works independently in finding a solution, i.e., evasive decisions are either made by a human (Ożoga and Montewka, 2018) or a machine (Liu et al., 2016), and the knowledge (intelligence) between the human and the machine is not shared.

1.3. Contributions

As presented above, limited studies focused on supporting the cooperation between humans and machines in conflict resolution for ships. Since the advantages of the human and the automatic systems are highly complementary, the combination will not only help the automatic systems gain knowledge from the human, supporting rule-compliant actions in the unmanned ship, but also help human on board take a correct operation in time in the manned ship.

In brief, supporting HMIs in decision making would combine the advantages of the human and the machine in preventing ship collisions, which is a necessary step for developing various MASS (Type I-III). However, few collision avoidance research considered HMIs in decision-making. Thus, the main research question for this study is: *how can a collision avoidance system that combines the wisdom of human and intelligence of machines in collision avoidance be developed?* The main contributions of this article is as follows:

1. We overview the HMIs levels of the existing CAS research and offer a clarification of HMIs in MASS with different autonomy levels;
2. We propose a Human-Machine Interaction oriented Collision Avoidance System (HMI-CAS) incorporating hybrid intelligence that combines the complementary strengths of human operators and artificial agents;
3. The proposed HMI-CAS visualizes the decision space of the machine and allows the human operators to intervene, which makes the decision-making process interpretable and interactive for human operators;
4. We take the under-actuated ship dynamics into consideration in the proposed HMI-CAS.

1.4. Outline

The structure of this article is addressed as follows. An overview of existing collision avoidance methods is presented in Section 2, followed by the framework of the HMI-CAS. Incorporating under-actuated ships has been shown in Section 4. Section 5 includes simulations to demonstrate the proposed HMI-CAS and Section 6 discusses the main findings

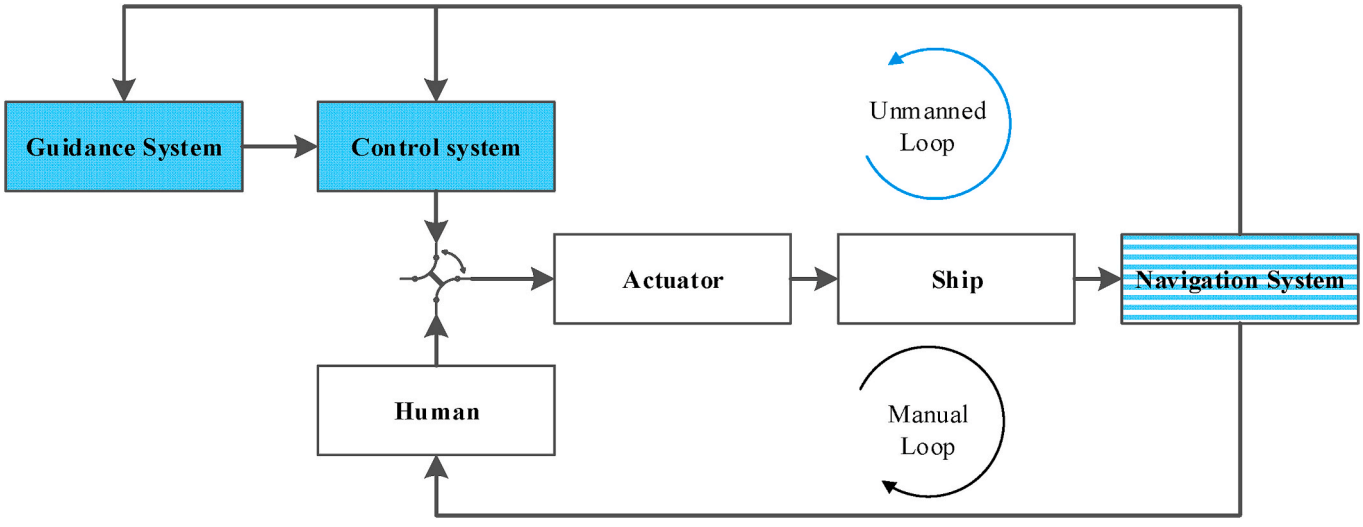


Fig. 1. Integration of the manned and unmanned control loops.

of this article. Conclusion and directions for future research are summarized in Section 7.

1.5. Literature review on ship collision avoidance

From existing Collision Avoidance Systems (CASs), two categorizes are observed. One focuses on developing alert systems for reminding human operators to detect dangers, while supporting the operator to find collision-free solutions is not a focus. The other one concentrates on developing the automatic system that avoids collisions automatically for MASS, while interacting with human operators are not considered. To our best knowledge, supporting HMIs in conflict resolution is rarely addressed in the literature (Huang et al., 2020a; Chen et al.).

1.6. HMIs in conflict resolutions

The interaction between human and machines in collision avoidance mainly refers to information exchanges. To see how the existing CASs support the interactions during conflicts, we firstly categorized the forms of outputs of various CASs, and then categorize HMIs according to the information flow between machine and human operators. Then, the feature of existing collision avoidance algorithms regarding the categorized forms of interactions is clarified.

The information delivering to human include, but not limited to.

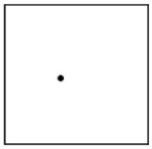
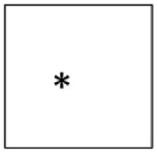
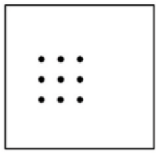
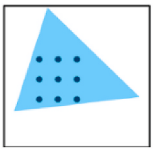
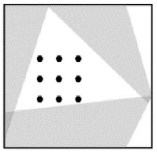
(1) One collision-free solution, noted as u;
(2) One best solution that minimizes the utility function, noted as u*;
(3) A finite number of safe solutions noted as $\mathbb{U}$;
(4) A closed region of collision-free solutions noted as $\mathscr{U}$;
(5) All closed regions of unsafe solutions noted as $\mathrm{K} = \bigcup_i \overline{\mathscr{U}}_i$.

Human operators can conduct the following operations using the above information:

a. Switching to manual mode when human operators prefer to steer the ship by themselves or the solution found is not fully satisfied.
b. Accepting the solution;
c. Changing the solution by modifying the utility function when the operator does not satisfy the selected "best" solution by machine;
d. Picking up one solution in given finite solutions $\mathbb{U}$ when the machine does not pick up any solution to human operators or the chosen one is not satisfied. (The machine shows several solution candidates);
e. Picking up one solution in a given collision-free set $\mathscr{U}$ when the machine does not pick up any solution to human operators or the chosen one is not satisfied. (The machine shows a set of feasible solutions);
f. Validating the safety of inputted solution when the machine does not pick up any solution or the chosen one is not satisfied. (The machine shows unsafe solutions)

Illustrations of the solution forms and relevant services are shown in Table 1. The solution space of the ships is presented, and each point in this space represents one maneuver. The black dot of Type (1) in Table 1 shows one type of information service that the machine delivers one feasible (but not necessarily optimal) collision-free solution to human operators. In Type (2), the "*" mark means the machine offers an optimal collision-free solution. In Type (3), the dots are the optional collision-free solutions that the machine can find. In Type (4), a subspace of solution space containing collision-free solutions is colored in blue. In Type (5), instead of finding a collision-free set, the machine identifies all the solution sub-spaces leading to the collision, colored in grey. In many cases, the combinations of these information types are needed to achieve more HMI types.

Table 1
Overview of different forms of solutions and relevant interactions.

Services from machine			Operations that human can do
Type (1)	A feasible solution: u.		a. Switch to manual mode; b. Accept the solution;
Type (2)	An optimal solution: u^*.		a. Switch to manual mode; b. Accept the solution; c. Modify utility function
Type (3)	Finite feasible solutions: $\mathbb{U}$.		a. Switch to manual mode; d. Pick up one solution in $\mathbb{U}$.
Type (4)	A closed region of feasible solutions: $\mathscr{U}$.		a. Switch to manual mode; e. Freely choose one solution in $\mathscr{U}$
Type (5)	All closed regions of dangerous solutions: **K**.		a. Switch to manual mode; e. Freely choose one solution in $\mathscr{U}$; f. Validate arbitrary solution inputted by human using K.

* Each dot in figures represents one control input to the ship. The region in blue is a collision-free sub-space, and the region in grey is a sub-space that leads to collisions.

1.7. Solution forms of existing algorithms

In (Huang et al., 2020a), the algorithms for conflict resolution have been categorized into 5 groups with 14 representative algorithms. In this section, the representative algorithms regarding the demand of HMIs addressed in Section 2.1 are compared and shown in Table 2.

"Rule-based" methods usually offer one feasible solution to operators or MASS controllers. The feasible solution could be a course (Naeem et al., 2012), a speed (Perera et al., 2012), or a pattern (enlarge rudder angle until it is collision-free) (Fang et al., 2017), etc. In these studies, Fuzzy logic based methods are popular (Perera et al., 2012; Wu et al., 2020). Nevertheless, since the collision check is not inclusive, some algorithms might not guarantee that the solution is collision-free.

"Virtual vector" methods offer a solution that might not be optimal but collision-free. The solution could be a collision-free course by Artificial Potential Field (APF) (Lyu and Yin, 2018) or a collision-free trajectory generated by Limited Cycle Method (LCM) (Mahini et al., 2013; Soltan et al., 2010).

"Discrete inputs" group follows a common idea that firstly discretizes the control space, and then finds the solution. Brute-force search checks the solutions from the discrete control space and returns one collision-free solution. Dynamic Window (DW) (Serigstad, 2017) and Decision Disc (DD) (Benjamin et al., 2006; Kuwata et al., 2014) enable to offer alternative solutions and an optimal solution. These algorithms search each feasible solution in solution space, and the optimal solution is found in these solutions. However, since the control space is discretized, these algorithms cannot offer a continuous solution space to users. Besides, the DW implies stopping the ship is always a safe option, and DD assumes the target-ship sailing with constant speed and course, which

Table 2
Overview of solution forms of collision avoidance algorithms.

	Algorithm	Encounter types	Output solutions					The physical meaning of the solution
			u	u^*	$\mathbb{U}$	$\mathscr{U}$	K	
Rule-based	Single-rule	Single-	◐	–	–	–	–	turning course (Naeem et al., 2012) or pattern (Fang et al., 2017)
	Multiple-rule	Single-	◐	–	–	–	–	heading & speed (Perera et al., 2012)
Virtual field	Article Potential Field	Multiple-	●	–	–	–	–	course/velocity (Lyu and Yin, 2017, 2018)
	Limited Cycle Method	Single-	●	–	–	–	–	trajectory(Soltan et al., 2009, 2010; Mahini et al., 2013)
Discrete inputs	Dynamic Window	Multiple-	●	●	●	◐	◐	speed & yaw rate (Loe, 2008; Serigstad, 2017)
	Decision Disc	Multiple-	●	●	●	◐	◐	velocity (Benjamin et al., 2006; Degre and Lefevre, 1981; Lenart, 1983; Pedersen et al., 2003; Kuwata et al., 2014; Szlapczynski and Krata, 2018)
	Discrete-Input Optimization	Multiple-	●	●	–	–	–	Velocity (Johansen et al., 2016) /rudder angle (Li et al., 2018)
	Lattice-Based Search	Multiple-	●	●	–	–	–	velocity (Shah et al., 2015; Švec et al., 2013)
	Brute-Force Search	Multiple-	●	–	–	–	–	rudder angle & operation time (Zhang et al., 2015)
Continuous inputs	Velocity Obstacle	Multiple-	●	●	●	●	●	Velocities (Zhuang et al., 2016)
	Vision Cone	Single-	●	–	–	–	–	Course (Wiig et al., 2017a, 2017b; Xue et al., 2011)
	MPC-Collision Avoidance	Multiple-	●	●	–	–	–	trajectory /desired forces (Chen et al., 2018; Ferranti et al., 2018; Abdelaal et al., 2018)
Re-planning	Fast Marching Method	Multiple-	●	●	–	–	–	Path (Liu et al., 2017)
	Projected Obstacle Area	Multiple-	●	●	–	–	–	Path (Larson et al., 2006, Zhao-Lin, 1988)

* Note: "●" means this algorithm matches the description; "◐" means this algorithm is not fully matching the description; "-" means this algorithm is not matching.

are not always the case at sea. Discrete-Inputs Optimization and Lattice-based search contain an optimization finding one collision-free solution. Thus, they can directly offer an optimal solution to users.

"Continuous inputs" methods search collision-free solutions in a continuous space. Both Visual Cone (VC) and Model Predictive Control based Collision Avoidance (MPC-CA) can find one collision-free solution, while MPC-CA could offer a solution to minimize a predefined cost function. Velocity Obstacle (VO) algorithm firstly identify the unsafe region in solution space (i.e., K) and then find an optimal solution in the complimentary space (i.e., $\overline{K}$).

Two representative algorithms of the "Re-planning" group are Fast Marching Method (FMM) and Projected Obstacle Area (POA). FMM firstly assigns a cost map and then finds an optimal path; POA assigns a prohibit region around the predicted position of obstacle and finds one path to avoid the prohibited region.

From Table 2, one can observe that different methods can provide different information and support relevant HMIs. It is worth noting that only VO algorithms can provide various types of information listed in Section 2.1 (i.e., (1)–(5)) and then can support all types of HMIs (âf).

1.8. A generic framework of HMI-CAS

As mentioned in Section 1 and 2, HMI for MASS with different autonomy levels requires different types of information. Thus, to develop the CAS supporting HMIs, the HMI-CAS needs to offer various forms of solutions, i.e., one feasible solution, one optimal solution, several feasible solutions, feasible ranges of solutions, and dangerous solutions.

1.9. Focuses of HMI-CAS

Different from the hybrid system presented in Fig. 1, this manuscript aims at adding an interaction between the automatic system and human, specifically, between the "Guidance" system and human operators (the red arrows shown in Fig. 2). "Guidance" system provides information to human, and human sends commands to "Guidance" system, e.g., authorization, intervention, etc. The CAS and an interface are composing of the collision avoidance module in the "Guidance" system. The entire GNC system with the designed CAS and interface is named as Human-Machine Interaction oriented Collision Avoidance System (HMI-

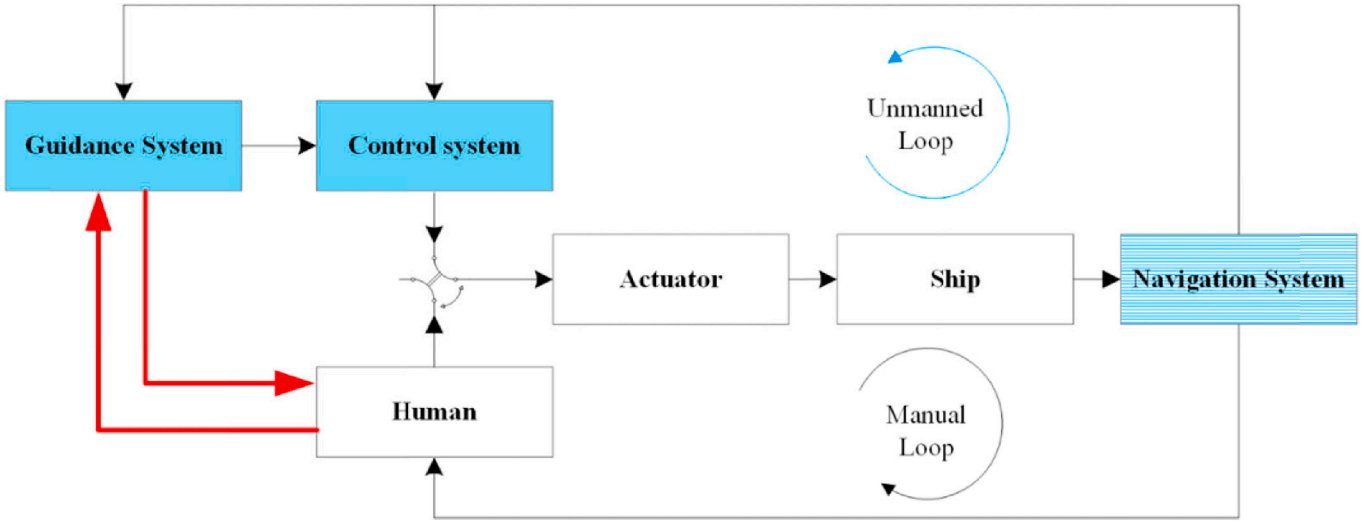


Fig. 2. Representation of control loop within the proposed HMI-CAS.

CAS).

To incarnate the connection between human operators and the "Guidance" system, an interface is introduced, that supports information exchange. An illustration of this idea is shown in Fig. 3. The blue arrow shows the information flow in the CAS, which delivers information to human operators. The green arrows represent the inputted information by humans, which feeds the human's decision backs to the machine and influences the behavior of the ship.

1.10. A generic framework of HMI-CAS

An abstract representation of HMI-CAS is shown in Fig. 4. The proposed HMI-CAS has two modes: manual mode and autonomous mode.

When there is no human intervention, the own-ship follows the waypoints and automatically takes evasive actions, i.e., autonomous mode. The framework of this system based on the GNC system. Three basic systems are inclusive: the Guidance system, the Navigation system, and the Control system.

The Navigation system contains sensors and transceivers that allow the ship to sense the surrounding environment and to exchange its state with other ships. The obtained information is inputted to the Guidance system. The Guidance system is performing as a "brain" that makes decisions according to obtained information. Moreover, this electronic brain is allowed to interact with humans via the designed interface. The Control system contains a controller that implements the input solution from the Guidance system and output relevant command to actuators, i.e., propeller and rudder.

To support interaction between human operators and the GNC system, an interface is added to the Guidance system. Thus, three main modules are included in the Guidance system, i.e., Global Planner, Local Planner, and Interface. The structure of each module is shown in Fig. 5.

Global Planner generates a planned waypoint for the system which guides the ship to the destination, noted as *wp*. This waypoint is generated by path planning algorithms or input by human operators.

Local Planner generates a collision-free solution considering the observed information, planned waypoints, and human's feedbacks, etc., noted as $\mathbf{u}^*$.

Interface has two main functions. For one, it presents information for human operators, e.g., a set of solutions leading to collisions, a set of collision-free solutions, and a selected solution to human operators; For the other, it collects orders from human and sends it back to local motion planner system. The orders could be an alternative solution, a command to stop/continue the existing mode, etc.

2. Global planner

Global Planner contains a path planning module in which various deliberate algorithms can be used, e.g., A* (Liu et al., 2019), Theta* (Kim et al., 2014), Dijkstra, Fast marching method (Liu and Bucknall, 2016), Potential Field (Xue et al., 2009), etc. An overview of these algorithms is shown in (Campbell et al., 2012). The basis of these methods is a known roadmap. Some other algorithms need an inputted cost function that is minimized to find the path. Alternatively, instead of using path planning algorithms, human operators also can directly input a desired path. Given the input/optimized path, the global planner also needs to find out which waypoint is activated, i.e., the waypoint that the ship is heading to.

3. Local Planner

Local Planner is the focus of the guidance system, which is in charge of finding a collision-free solution. It consists of three key processes, i.e., "trajectory prediction", "conflict detection", and "conflict resolution". The "trajectory prediction" outputs the trajectories of the OS and the TS, i.e., $\mathbf{x}_i(t)$ and $\mathbf{x}_j(t)$. Based on these predicted outcomes, the "conflict detection" calculates the relevant collision risk and triggers an alarm and "conflict resolution" if necessary. Lastly, the conflict resolution finds a collision-free solution according to input information from "trajectory prediction", "conflict detection", and feedbacks from a human. The outputted information is delivered to the "Interface" module and the "Control" system. If no command from the "Interface", the optimal solution will be executed by the "Control" system.

4. Interface

"Interface" is designed for the interactions between human operators and the machine, which shows the supporting information for collision avoidance and collects feedback from human operators.

The information from the machine includes collision-free solutions, best solutions, dangerous solutions, etc. Since the input information

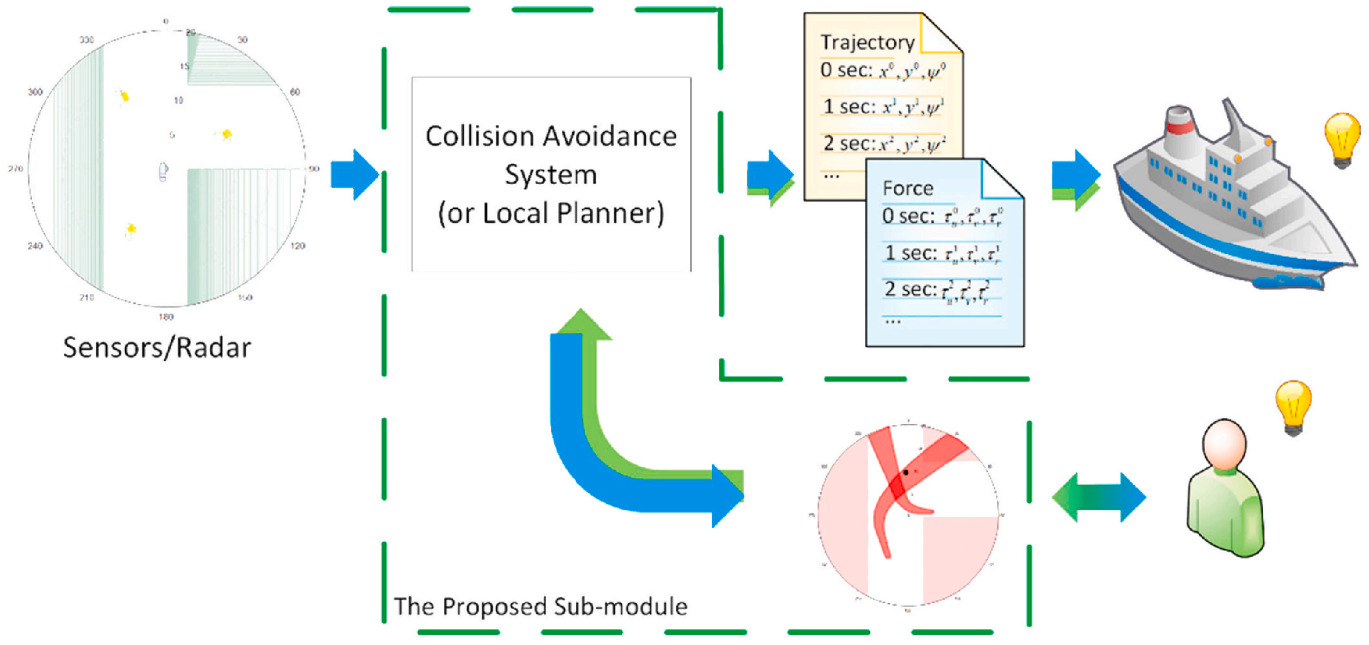


Fig. 3. Illustration of the proposed HMI-CAS.

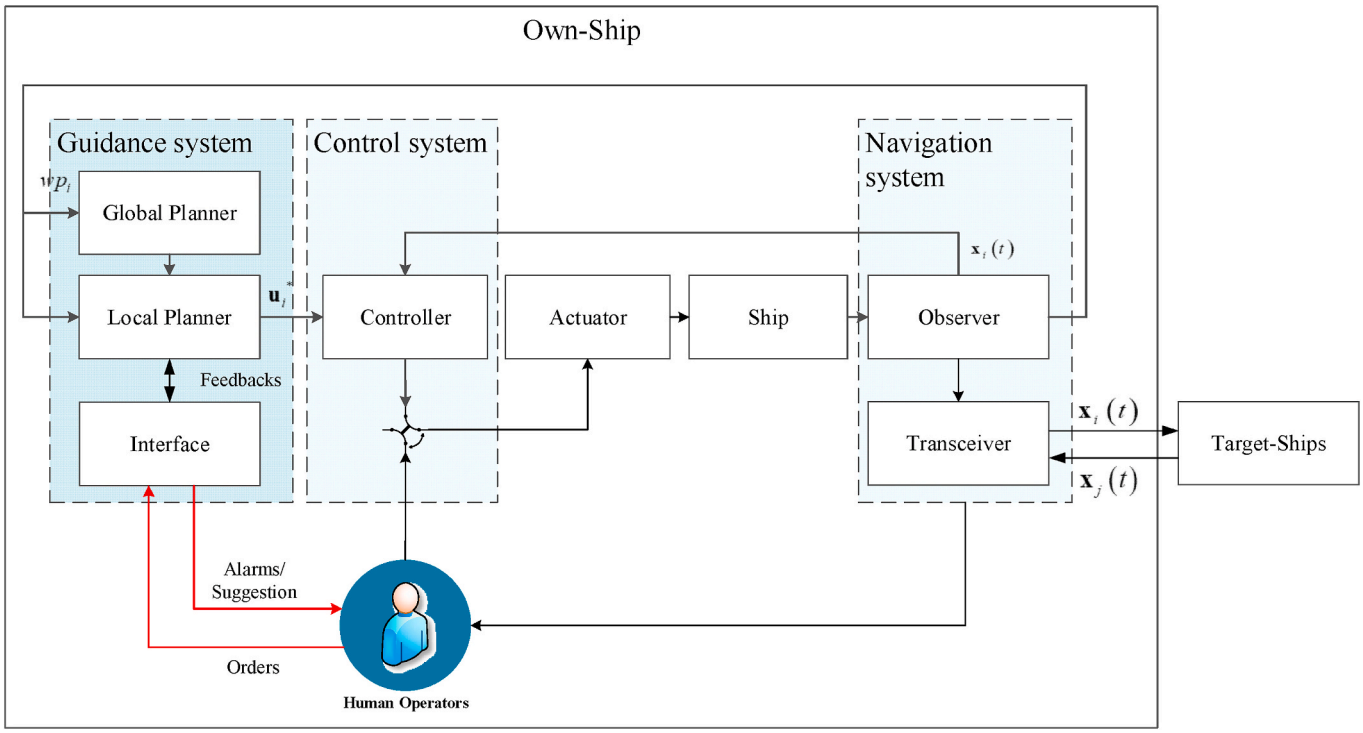


Fig. 4. Abstract representation of the proposed HMI-CAS. note: sub-script i ($\bullet_i$) represents the states related to the own-ship; j ($\bullet_j$) is related to the target-ship.

from conflict resolution might not be directly readable for human operators, we design a mapping sub-module that presents the resolution space. This map, together with a solution selected by the machine, is projected in the human-machine interface. The human operator can then read the map to understand the selected solution of the machine, authorize the system to continue, intervene in the system, or find a new solution if necessary.

The information from human operators to the machine is the new solution they input in the interface, noted as $\mathbf{u}^{\text{human}}$. This new solution is sent back to the mapping sub-module that translates the solution to the machine's language.

4.1. Requirements on HMI-CAS for achieving HMIs

Although the manual mode is kept, the function for the designed HMI-CAS is to show the decision process of the machine, providing the opportunity for human to intervene in the decisions. In this way, the operator can build trust with this new system, and then the operator does not need to stay onboard for directly controlling the ship. The human operators, then, can intervene/interact with the machine in offshore control centers. In brief, the HMI-CAS can be applied to MASS with different autonomy levels where human is involving. This system supports human in the following aspects:

(1) to identify who have conflicts with the own-ship;
(2) to notice whether it is necessary to take actions;
(3) to show how does the CAS system avoid the dangers;
(4) to be aware of what kind of operations are dangerous/safe;
(5) to inform whether the chosen solution by the human is safe/unsafe;
(6) to intervene in automated collision avoidance if necessary.

To meet these demands, the "Interface" in HMI-CAS needs to be user-friendly. In particular, the interface facilitates the users to read the information, to understand the collision-free solutions selected by the machine, and to intervene in the conflict resolution.

Moreover, this system needs to support various types of cooperation in conflict resolution. Thus the "Conflict Resolution" in the "Local Planner" module needs to offer various types of solutions for users. For instance, one collision-free solution, one best solution, a set of safe/unsafe solutions, etc.

From the comparison in Section 2.2, we find that the VO algorithm meets our demands: firstly, the VO algorithm presents a set of dangerous solutions (i.e., K) that allows the human operators to validate their solutions (safe or not); secondly, the optimization in the complementary of K set is possible. These features can satisfy the demand of human-machine cooperation in the proposed HMI-CAS. Moreover, the form of the solution of VO algorithms is friendly for human operators to read and to implement.

Therefore, in this article, a family of VO algorithms is adopted for developing the proposed HMI-CAS. When applying the VO algorithms in the maritime domain, this family of algorithms has some limitations. Firstly, the VO algorithm assumes the target ship sail with constant speed and course which is noted as semi-dynamic TS. Secondly, the dynamics of the ship are ignored in finding a collision-free solution. Some modifications are made to overcome the limitations. In (Huang et al., 2018), researchers introduced a non-linear VO algorithm and a probabilistic VO to deal with semi-dynamic TS. In (Huang et al., 2019), the authors incorporate the ship dynamics by using the Generalized VO (GVO) algorithm, in which the ship is fully actuated. In this article, we employed the GVO algorithm to achieve the proposed HMI-CAS and consider under-actuated ships that are more common in practice.

HMI-CAS considering under-actuated ship dynamics.

4.2. Under-actuated ship dynamics

In general, the ship is an under-actuated vehicle with two types of control inputs, i.e., thrust (T) and torque (N). The surge speed is controlled by thrust and the yaw is determined by torque, while the sway speed is not directly influenced by control inputs. A representation of such ship dynamics can be formulated as follows:

$$\begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{M}_{3\times3} \end{bmatrix} \dot{\mathbf{x}} = [\mathbf{R}(\psi)\mathbf{v} - \mathbf{C}(\mathbf{v})\mathbf{v} - \mathbf{D}(\mathbf{v})\mathbf{v}] + \mathbf{B}_{6\times2}\boldsymbol{\tau} \tag{1}$$

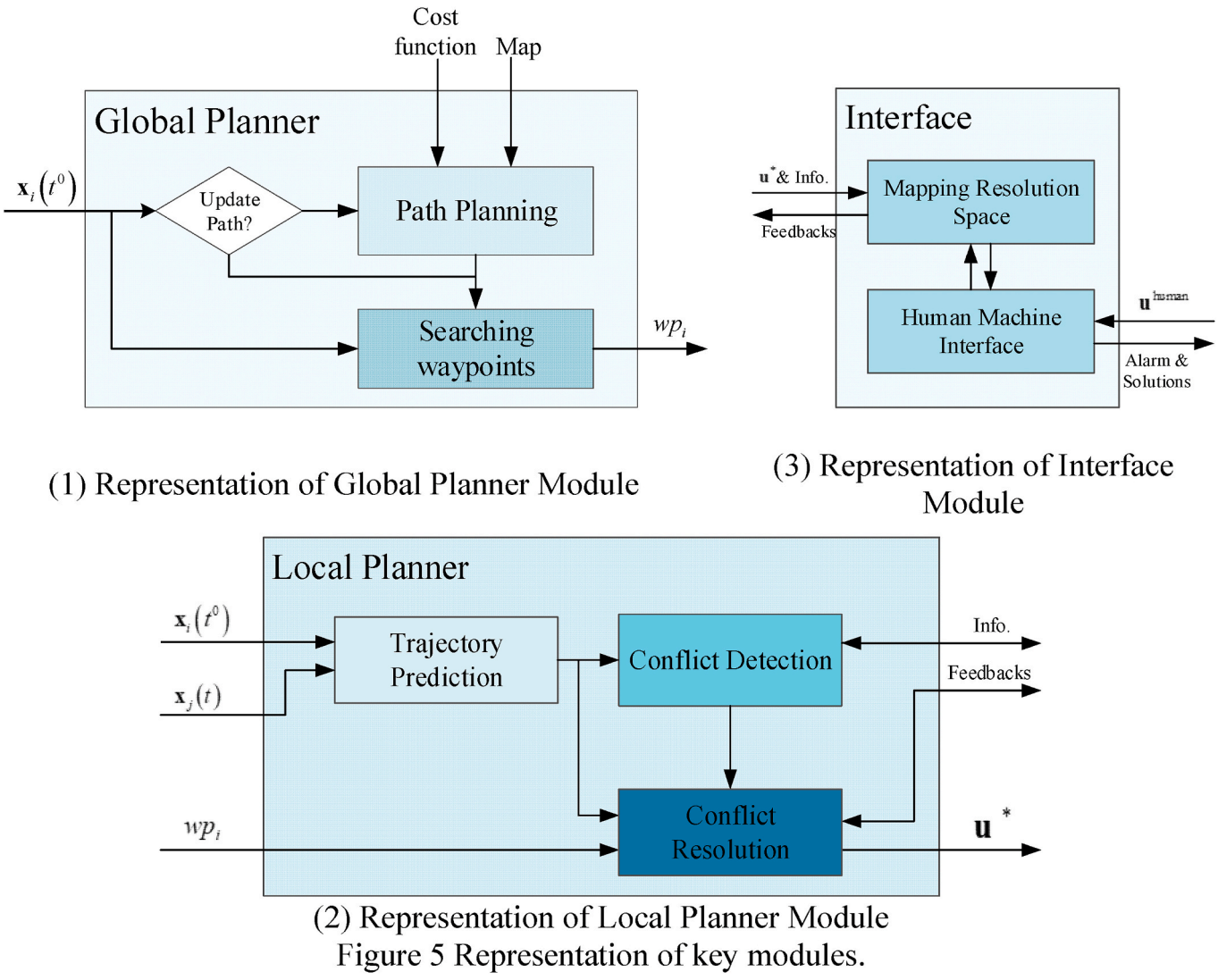


Fig. 5. Representation of key modules.

where $\mathbf{x}$ is the system state containing coordinates (x,y), heading angle ψ, and $\mathbf{v}$ that consists of linear velocities (u,v) and angular velocity r. $\boldsymbol{\tau}$ is inputs vector with thrust T and torque N. $\mathbf{M}$, $\mathbf{C(v)}$, $\mathbf{D(v)}$, and $\mathbf{R}(\psi)$ are inertia matrix, Coriolis–centripetal matrix, damping matrix, and rotation matrix, respectively. Moreover, $\mathbf{B}_{6\times 2} = \begin{bmatrix} \mathbf{0}_{2\times 3}, & \begin{matrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{matrix} \end{bmatrix}^{\mathrm{T}}$. Details refer to (Fossen, 2002). Illustration of the coordination system is presented in Fig. 6.

Moreover, we add a PD controller as a high-level controller and switch the control input from the force $\boldsymbol{\tau}$ to the desired velocity $\mathbf{u}$. The PD controller is formulated as:

$$\boldsymbol{\tau} = \mathrm{K_p}(\mathbf{u} - g(\mathbf{x})) - \mathrm{K_d}\dot{g}(\mathbf{x}) = -\mathrm{K_p}g(\mathbf{x}) - \mathrm{K_d}\dot{g}(\mathbf{x}) + \mathrm{K_p}\mathbf{u} \tag{2}$$

where, $g(x)$ is an observed function, $\mathrm{K_p}$ and $\mathrm{K_d}$ are feedback gains. Let formulate the observe function as

$$g(x) = V \text{ and } \mathbf{x}\dot{g}(x) = V\dot{\mathbf{x}} \tag{3}$$

Then, the new system with the PD controller uses the desired velocity

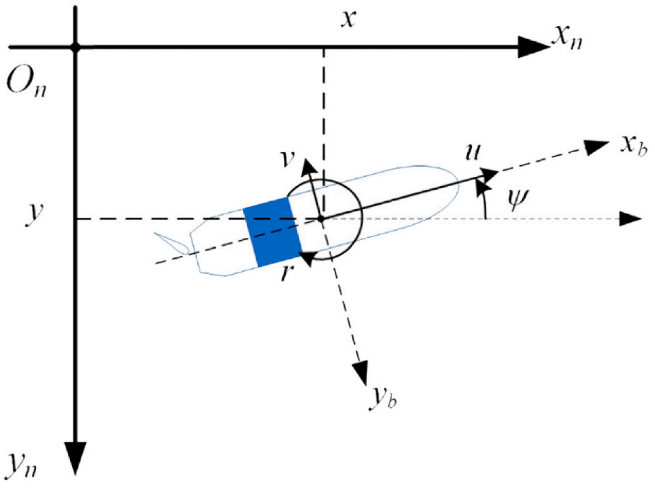


Fig. 6. The inertial frame $\{n\}$ and the body frame.$\{b\}$

$$= \left(\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix} + \mathrm{BK_dV} \right)^{-1} [\mathbf{R}(\psi(t))\mathbf{v}(t) - \mathbf{C}(\mathbf{v}(t))\mathbf{v}(t) - \mathbf{D}(\mathbf{v}(t))\mathbf{v}(t) - \mathrm{K_pV}\mathbf{x}(t)] + \left(\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix} + \mathrm{BK_dV} \right)^{-1} \mathrm{BK_p}\mathbf{u}(t) = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{4}$$

$$= \left(\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix} + \mathrm{BK_dV} \right)^{-1} [\mathbf{R}(\psi(t))\mathbf{v}(t) - \mathbf{C}(\mathbf{v}(t))\mathbf{v}(t) - \mathbf{D}(\mathbf{v}(t))\mathbf{v}(t) - \mathrm{K_pV}\mathbf{x}(t)] + \left(\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix} + \mathrm{BK_dV} \right)^{-1} \mathrm{BK_p}\mathbf{u}(t) = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{5}$$

as inputs (i.e., $\mathbf{u}$):

Since equations (4) and (5) is non-linear, we can approximate the state of the ship with the help of Runge-Kutta (RK) Integration and Taylor expansion law. Specifically, we can formulate the position of the ship at time t via the changes of the desired velocity, $\Delta\mathbf{u} = \mathbf{u} - \mathbf{u}(0)$.

$$\mathbf{x}(t) \approx \int_0^t \mathbf{f}(\mathbf{x}^0, \mathbf{u}^0)d\tau + \int_0^t \Delta\dot{\mathbf{x}}(\tau)d\tau = \tilde{\mathbf{x}}(t) + G(t)\Delta\mathbf{u} \tag{6}$$

where $G(t) = \int_0^t e^{A(t-\tau)}Bd\tau$(with $A = \mathbf{f_x}$and$B = \mathbf{f_u}$) and $\tilde{\mathbf{x}}$ is the estimated state of the ship calculated via RK method with a known initial state $\mathbf{x}^0$ and a known initial input $\mathbf{u}^0$.

4.3. GVO algorithm considering under-actuated dynamics

Given Equation (6), the position of the ship at time t is formulated as:

$$\mathbf{P}_i(t) = C\tilde{\mathbf{x}}_i(t) + CG(t)\Delta\mathbf{u}_i \tag{7}$$

where $C = [\mathbf{I}^{2\times2}, \mathbf{0}^{2\times4}]$ contains a 2-by-2 identical matrix and a 2-by-4 zero matrix. Thus, the necessary condition of collision at time t is presented as:

$$\mathbf{P}_i(t) \in \mathbf{P}_j(t) \oplus ConfP \tag{8}$$

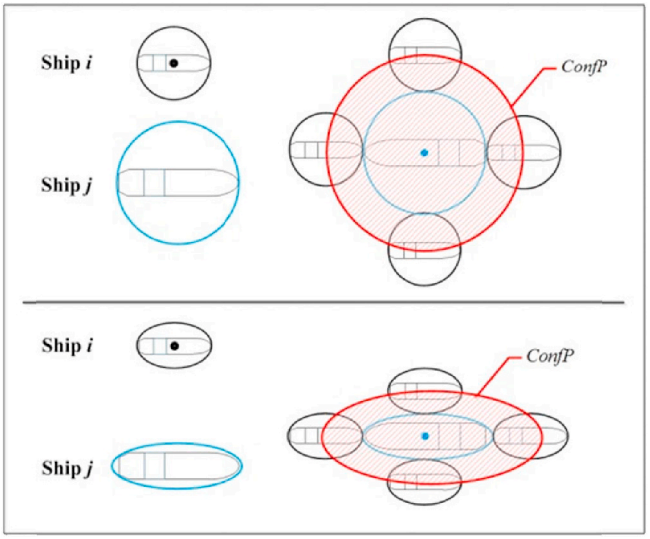


Fig. 7. Two representations of *ConfP* (Huang et al., 2019).

ConfP is the adjacent safety region surrounding the target ship. If the ship is represented as a circle, the *ConfP* is a set of positions that lead to the overlap of two circles, i.e. an enlarged circle, as shown in the upper panel of Fig. 7. If the ship is formulated as an ellipse, the *ConfP* would shape as an enlarged ellipse, as shown in the lower panel of Fig. 7. In this article, the circular-shaped ship is accepted and circular *ConfP* is used.

Equation (8) is the condition of a collision, which means the OS violates the safety region of the TS. By solving Equation (8), a sub-set of changes of the desired velocity leading to a collision at time t is collected, i.e.,

$$\Delta\mathbf{u}_i \in (CG(t))^{-1} \cdot \left[-\left(C\tilde{\mathbf{x}}_i(t) - \mathbf{P}_j(t)\right) \oplus ConfP \right] = \mathrm{sUO}(t) \tag{9}$$

Then, the changes in the desired velocity leading to a collision are the union of subsets, i.e.,

$$\mathrm{UO} = \bigcup_t^{\infty} \mathrm{sUO}(t) \tag{10}$$

4.4. Under-actuated ship with a PD controller

There are two possible choices to handle the changes from full-actuated dynamics to under-actuated dynamics.

The first choice is releasing the constraint on sway direction and asking the ship to track the desired surge speed $\breve{u}$ and desired heading$\breve{\psi}$. Thus, $\mathbf{u}$ consists of surge speed and heading. Since this system is stable, the sway speed will converge to 0 with time goes infinite. That implies the following settings:

$$\mathbf{u} = \begin{bmatrix} \breve{u} \\ \breve{\psi} \end{bmatrix} \tag{11}$$

$$g(\mathbf{x}) = V\mathbf{x} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \mathbf{x} \tag{12}$$

$$\dot{g}(\mathbf{x}) = V\dot{\mathbf{x}} \tag{13}$$

Substitute Equation 11–13 back to Equations 5–10, then the UO set of $\mathbf{u}$ leading to a collision is collected.

The second choice sets $\mathbf{u}$ as a combination of resultant speed U and course χ, i.e., we have:

$$\mathbf{u} = \begin{bmatrix} U \\ \chi \end{bmatrix} \tag{14}$$

$$g(\mathbf{x}) = \begin{bmatrix} \sqrt{u^2 + v^2} \\ \psi - \mathrm{atan}(v/u) \end{bmatrix} \tag{15}$$

$$\dot{g}(\mathbf{x}) = \begin{bmatrix} 0 & 0 & 0 & \dfrac{u}{\sqrt{u^2+v^2}} & \dfrac{v}{\sqrt{u^2+v^2}} & 0 \\ 0 & 0 & 1 & \dfrac{v}{\sqrt{u^2+v^2}} & \dfrac{-u}{\sqrt{u^2+v^2}} & 0 \end{bmatrix} \dot{\mathbf{x}} = V\dot{\mathbf{x}} \tag{16}$$

Substitute (14)–(16) back to (4), (6)–(10), then the resultant speeds

and courses leading to a collision are collected in the UO sets.

4.5. Optimization for auto mode

The Auto mode of the HMI-CAS neglects the regulations. The task of finding rule-compliant actions is assigned to human operators who are experts in understanding rules. The HMI-CAS only offers an optimal solution to human operators regarding cost function defined in Equation (19).

The rules for finding a new desired velocity in HMI-CAS is changed as follows, and they are presented in order of priority. Prior to that, some symbols are explained. Reference velocity is noted as $\mathbf{r}$, the origin of the space is noted as $\boldsymbol{O}$ that represents the desired velocity in the last control loop ($\mathbf{u}^0$), and $\mathbf{u}^*$ is the optimal desired velocity in this control loop.

Rule 1. if $\Delta\mathbf{r} = (\mathbf{r} - \mathbf{u}^0) \notin \mathrm{UO}$, the OS is expected to choose $\mathbf{r}$, i.e., $\mathbf{u}^* = \mathbf{r}$.

Rule 2. if $O \notin \mathrm{UO}$, the OS prefers to continue with its initial desired velocity, $\mathbf{u}^* = \mathbf{u}^0$,

Rule 3. a new $\mathbf{u}^*$ is close to its current velocity, satisfying $\Delta\mathbf{u} = (\mathbf{u}^* - \mathbf{u}^0) \notin \mathrm{UO}$.

When the reference velocity and the initial desired velocity are both unsafe, finding a new solution is necessary, i.e. Rule 3 is activated. This problem can be modeled as a non-convex optimization problem since the solution space $\mathbb{R}^2\backslash\mathrm{UO_{ij}}$ is non-convex.

The final control input for ship i not only needs to fall in $\mathbb{R}^2\backslash\mathrm{UO_{ij}}$, but also satisfy the kinematic constraints, i.e., $u \in [u_{\min}, u_{\max}]$ and $\psi \in [-\psi_{\max}, \psi_{\max}]$, noted as U_i^{bound}. Thus, the feasible space for ship i to prevent collision with ship j is $U_{ij}^{fea} = \overline{\mathrm{UO_{ij}}} \cap U_i^{bound}$. An illustration has been shown in Fig. 8 (1). Obviously, U_{ij}^{fea} is non-convex, and an approximation of feasible space is necessary.

i. Approximation of feasible space.

Firstly, a convex hull is employed to approximate the intersection of $\mathrm{UO_{ij}}$ and U_i^{bound}, noted as$\mathrm{CH}(\mathrm{UO_{ij}} \cap U_i^{bound})$, see Fig. 8 (2). Secondly, the closest point on the boundary of $\mathrm{CH}(\mathrm{UO_{ij}} \cap U_i^{bound})$ to the initial desired velocity $\mathbf{u}^0$ is found, which is $\mathbf{w}$.

$$\mathbf{w} = \underset{\mathbf{u} \in \partial\mathrm{CH}\left(\mathrm{UO_{ij}} \cap U_i^{bound}\right)}{\operatorname{argmin}} \left\|\mathbf{u} - \mathbf{u}^0\right\| \tag{17}$$

where ∂ refers to the boundary of a set. In (Alonso-Mora et al., 2018), the authors reported when the target velocity is on the boundary of the VO set, two vehicles will approach each other infinitely close. Thus, in this system, a repulsive term $\hat{\mathbf{w}}$is adopted with $\mathbf{w} := \mathbf{w} + \mathbf{w}/\|\mathbf{w}\|\hat{\mathbf{w}}$ and $\hat{\mathbf{w}} = 0.02$, as shown in Fig. 8 (3). Lastly, U_{ij}^{fea} is approximated via a linear constraint that is formulated as:

$$\tilde{U}_{ij}^{fea} = \begin{cases} \{\mathbf{u} | (\mathbf{u} - \mathbf{w}) \cdot (\mathbf{w} - \mathbf{u}^0) \geq 0\} \text{ if } \mathbf{u}^0 \in \mathrm{CH}\left(\mathrm{UO_{ij}} \cap U_i^{bound}\right) \\ \{\mathbf{u} | (\mathbf{u} - \mathbf{w}) \cdot (\mathbf{w} - \mathbf{u}^0) \leq 0\} \text{ otherwise} \end{cases} \tag{18}$$

In return, the feasible space becomes convex (see Fig. 8 (3)).

ii. Formulation of the optimization problem.

Based on the approximated feasible space, the collision problem can be formulated as a quadratic problem:

$$\text{Minimize } J_{UO}(\mathbf{u}_i) = \left(\mathbf{u}_i - \mathbf{u}_i^0\right)^T \begin{bmatrix} a_u & 0 \\ 0 & a_\psi \end{bmatrix} \left(\mathbf{u}_i - \mathbf{u}_i^0\right) \text{Subject to: } \mathbf{u}_i \in \bigcap_{j \neq i} \tilde{U}_{ij}^{fea}, \text{ and } \mathbf{u}_i \in U_i^{bound} \tag{19}$$

The matrix diagonal matrix is introduced to reflect the experience in ship navigation that course turning is more popular in collision avoidance at sea, i.e. $a_u >> a_\psi$.

4.6. Case studies

In this section, a two-ship scenario is simulated to demonstrate the proposed HMI-CAS in different modes, i.e., Auto mode (Section 5.2.1) and Manual mode (Section 5.2.2). Moreover, to show the performance of the HMI-CAS in multiple ships, Section 5.2.3 is presented, where the ships are all installed the HMI-CAS and performing in Auto mode.

4.7. Setups

In simulations, CyberShip II is chosen as a basic ship model. The scale factor of the ship is 1/70. The length of the ship in the real-world is 87.85 [m]. The safety distance of the ship in the real-world is set as 0.2 [NM] which is 4 times of the ship length. Without a special explanation, all the parameters in simulations are in the real world. To obtain the parameters in the scaled-world, the Froude scaling law is used. Read more in the appendix in paper (Huang et al., 2019).

The feedback gains of ships are $\mathrm{K_p}$ = diag(100, 200, 10), and $\mathrm{K_d}$ = diag(5, 5, 5) for fully actuated ship and $\mathrm{K_p}$ = diag(100, 10), and $\mathrm{K_d}$ = diag(5, 5) for under-actuated ship. The course turning has priority for ship avoiding collisions, thus, a_u = 81 and a_ψ = 1. The ConfP is approximated by its circumscribed polygon, i.e. a 24-side polygon. The

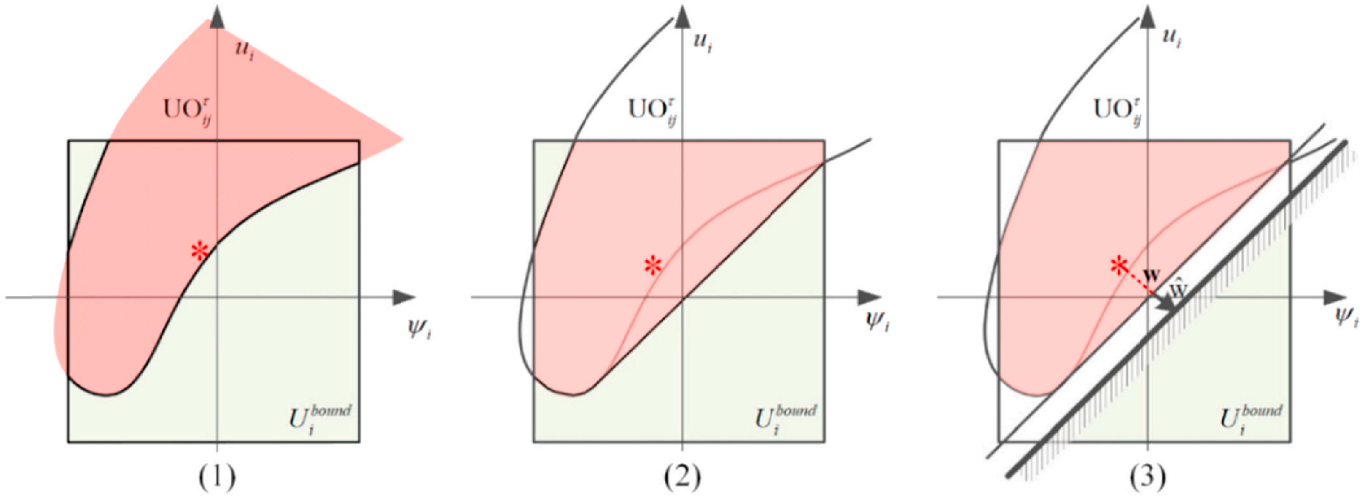


Fig. 8. Illustration of approximation of feasible space U_{ij}^{fea}.

simulator is designed on Matlab R2018b.

Both the two-ship encounter and multi-ship encounter are considered. In the two-ship encounter scenario, the ship under control is named as own-ship (OS). The OS is located at the origin of the coordination, i.e., (0,0) and sails to the North with 10 [knots]. The encountering ship labeled "NO. 2" is the target ship (TS). The TS sails to the Southwest with a speed of 10 [knot]. In this scenario, the TS is sail with a constant speed and course since the TS is the "Stand-on" ship according to COLREGs.

In the multiple-encounter scenario, the ship located at the origin is the OS and other ships are placed around the OS. The layout of these ships are shown in Fig. 12 (1).

4.8. Two-ship scenarios

Setting of two-ship encounter scenario using the HMI-CAS.

In this scenario, the OS is controlled by HMI-CAS, while human operators just monitor the ship. To see the difference due to different dynamics and different tracking modes, three groups of experiments are designed.

- Standard Group (SG) employs the fully actuated ship, which uses the same setting in (Huang et al., 2019).
- Control Group 1 (CG1) uses the under-actuated ship and tracks the surge speed and heading, while the sway speed is uncontrollable.
- Control Group 2 (CG2) also uses the same under-actuated ship in CG1 but tracks the resultant speed and course instead of surge speed and heading.

The layout of the encounter and the initial UO space of the OS in these three groups are presented in panel (2)–(4) in Fig. 9. In these panels, the horizontal axis indicates changes on the desired heading, and

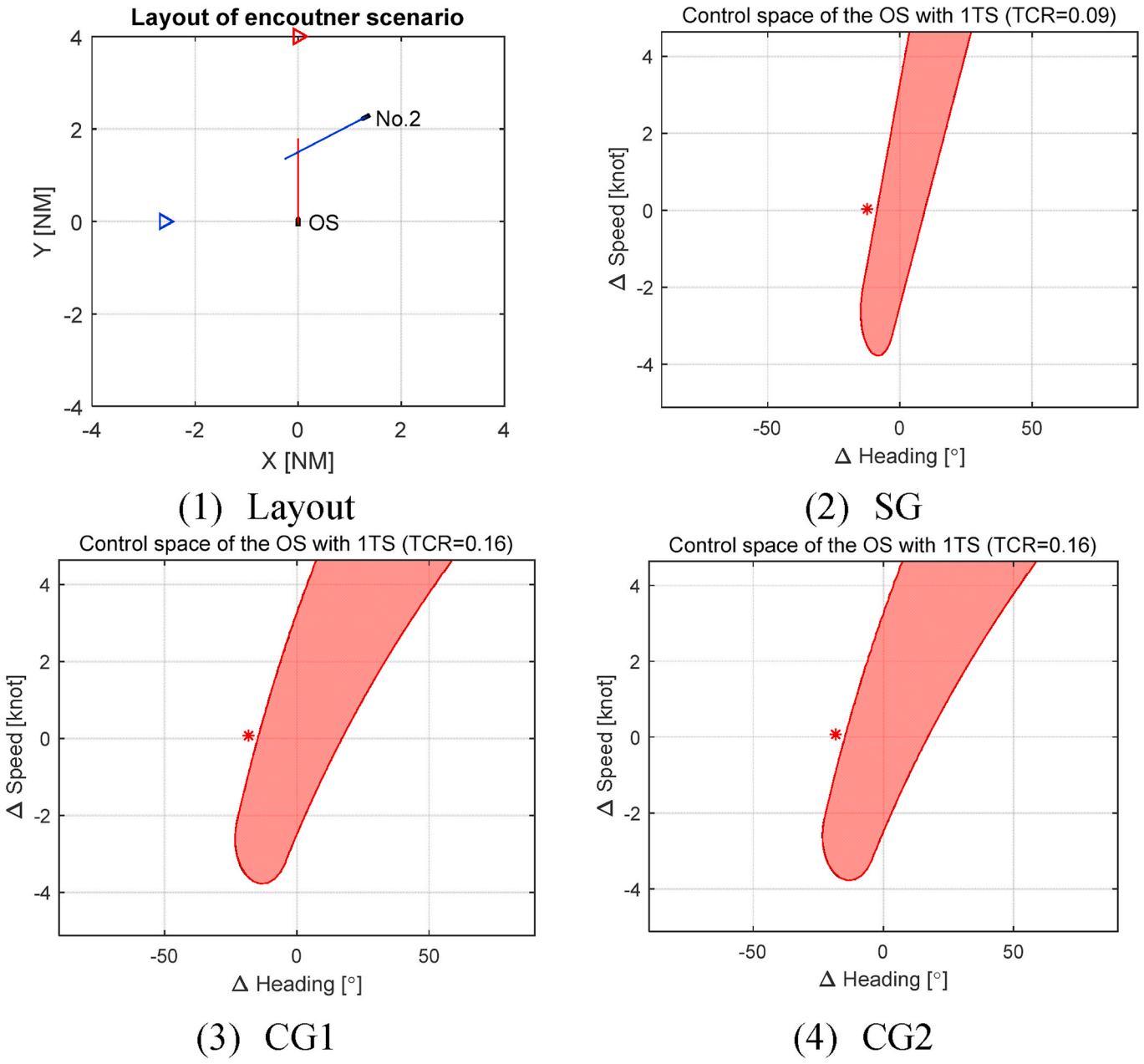


Fig. 9. Layout of encounter and control space of the OS in different groups of experiments. (TCR is time-varying collision risk measure defined in (Huang and Van Gelder, 2020)).

the vertical axis shows the changes on the desired speed. The negative directions refer to the port-side turning and deceleration, while the positive directions indicate the starboard-side turning and acceleration. In each panel, the red region represents the desired velocity leading to a collision with one obstacle, i.e., ship No. 2, while the blank region is collision-free. The red asterisk in these panels shows the optimal solution that HMI-CAS found, which suggest port-side turnings.

The control space, then, is presented to human operators via the Interface in Section 3, and the operators can gain the following information/services:

(1) Ship No. 2 is dangerous for the OS since the existing velocity is inside of the red region generated by Ship No. 2;
(2) It is necessary to take actions since the existing velocity is inside of the red region;
(3) The desired velocity inside of the reg region is dangerous, while arbitrary velocity out of the red region is safe;
(4) The red asterisk in the control space is the suggested optimal solution by HMI-CAS and without intervention from human operators, this solution will be executed by the system; (**Auto Mode**)
(5) When the operator disagrees with the suggested solution, the operator can choose another solution from the blank region and feedback it to the system. (**Manual Mode**)
(6) When the operator shows the velocity inside the red region, an alarm is triggered for reminding the operators that the chosen velocity is unsafe;

Two-ship encounter scenario using the Auto mode of the HMI-CAS.

Since the collision-free control space is non-convex, a linearization of the feasible control space is needed. The result of the optimization indicates that the port-side turn is optimal, i.e., the machine suggests port-side turns to avoid the collision. Subsequently, the OS is increasing its heading (for SG and CG1) or its course (for CG2) to prevent the collision. The behaviors of the ship in succeeding time are recorded in Fig. 10.

In Fig. 10, panel (a) shows the trajectories of the ships, and panel (b) is their relative distance over time. The lines in red represent the result from SG, the lines in green refer to CG1, and those in red are from CG2. For panel (c)-(h), the upper panels show the surge speed, sway speed, resultant speed, and the desired speed. The lower panels show the course, heading, and the desired heading (or course).

From the relative distance over time, it is observed that all these groups can support the ship to avoid collision successfully. However, the process of collision avoidance is different. In SG, the OS is fully actuated and the sway speed is tracked to be 0 [m/s]. Thus, the trajectory of the OS (the line in black) is nearly straight with curvature when the ship is turning, see panel (a). However, the trajectories of the OS in CG1 and CG2 are more smooth, which are consists of curves. In CG1, the ship gives up the sway speed and tracks the desired surge speed and heading. Therefore, from panel (e) and (f), it is observed that the sway speed exceeds 3 [knots] which is 30% of the full surge speed. Moreover, since the sway speed also influences the heading of the ship, the overshooting of the heading is also observed. Panel (g) and (h) show that the states of the ship in CG2 in which the ship tracks the resultant speed and course, which perform better than CG1. Specifically, the sway speed is not exceeding 2 [knots] and the overshooting of course is also not obvious. Moreover, the CG2 has fewer oscillations. Read more in Section 6.3.

4.8.1. *Two-ship encounter scenario using the manual mode of the HMI-CAS*

In the Auto mode, the HMI-CAS can support the ship to avoid collisions, while, the evasive action might violate the COLREG rules. Then, a manual mode is activated to prevent violations.

In the two-ship encounter scenario, the OS is the “give-way” ship and the TS is the “stand-on” ship and the “give-way” ship in this encounter scenario is not allowed to take port-side turns. For this simple scenario, it is possible to add an additional rule to prevent the port-side turn in the Auto mode. However, for more complicated encounter scenarios, it is challenging since the regulations do not enumerate all rule-compliant actions. The interpretations of regulations and the judgments of the encounter situation are all depending on the experience of navigators which might differ from one to another.

To avoid rule violations in the complicated encounters, the HMI-CAS can be turned to the manual mode. The manual mode presents the solution space as in the auto mode, while human operators enable to interact with the system. Specifically, they can assign a new collision-free solution, that is out of UO sets, to the system. In this way, human knowledge is introduced into the decision-making process, especially in the understanding of rules.

Fig. 11 demonstrates this mode in three groups presented in Section 5.2.1., i.e., SG, CG1, and CG2. In these systems, the human operators intervene in the initial choice of the system and monitor the collision avoidance process in each group. Panel (a) presents the trajectories, and panel (b) provides the relative distance; panel (c) and (d) show the recorded states of the ship in SG; panel (e) and (f) present the states of the ship in CG1; panel (g) and (h) display the ship’s states in CG2.

When the starboard solution is the input from the human operators, the ship would not return to the port-side turn in the next time step since the port-side solution is no longer optimal. Similar to the results presented in Fig. 10, the SG with the fully actuated dynamic has the best performance. When the ship is underactuated, CG2 has less oscillation and overshooting (the lower panels in the third and the last column).

4.9. *Multiple-ship scenario*

In this section, all the ships have installed HMI-CASs and switched them to Auto mode. Moreover, to reduce the oscillation, the PD controller will track the course and resultant speed. This simulation is used to demonstrate the performance of HMI-CAS in multiple-encounter scenarios when the ship is under-actuated.

The ships are cooperative and find a collision-free solution according to their order. Specifically, the “NO. 1” ship would find its collision-free solution first and share its predicted trajectory with other ships; the “NO. 2” ship then finds its solution based on the updated trajectory of the “NO.1” ship and other ships’ trajectory; this process repeats for the succeeding ships. These settings give the back ships higher priorities than the front ships. In an extreme case, the front ships have found collision-free solutions and the last ship does not need to change its course and speed.

The layout of the encounter scenario is presented in the first panel in Fig. 12. All the ships head to a common point ([0, 1.5] NM) with speed 10 knots. Thus, 0.15 h later, all ships would collide at the common point if no evasive actions are taken.

Fig. 12 shows the vision from the “NO.1” ship at the beginning, where the panel (1) is the layout, (2) is the UO sets in control space, (3) shows relative distance (solid lines) and the predicted relative distance (dotted lines), (4) speeds and desired speeds, (5) course and desired course, and (6) 1 NM around the “NO.1” ship showing course, heading, and trajectories.

Since the origin of the panel (2) is in the UO sets, which implies the desired initial velocity and the reference velocity are both unsafe, the ship determines to find a new collision-free solution, i.e., Rule 3 is activated. From the panel, it is observed that the collision-free area (the area in the blank) is non-convex. Thus, the approximation of the collision-free space in Section 4.4 is introduced. Through an optimization process, an optimal collision-free solution is found which is reducing the resultant speed by 5 knots and turns to the starboard side (right-hand side). This strategy basically asks the ship to postpone the collision with the “NO. 4” ship.

Fig. 13 shows the encounter-scenario after 16 s. The “NO.1” ship detects that the initial desired velocity is too close to the UO set, which might result in the collision with the “NO.4” ship. Thus, the ship activates the HMI-CAS to find a new collision-free solution. In (2) panel, a

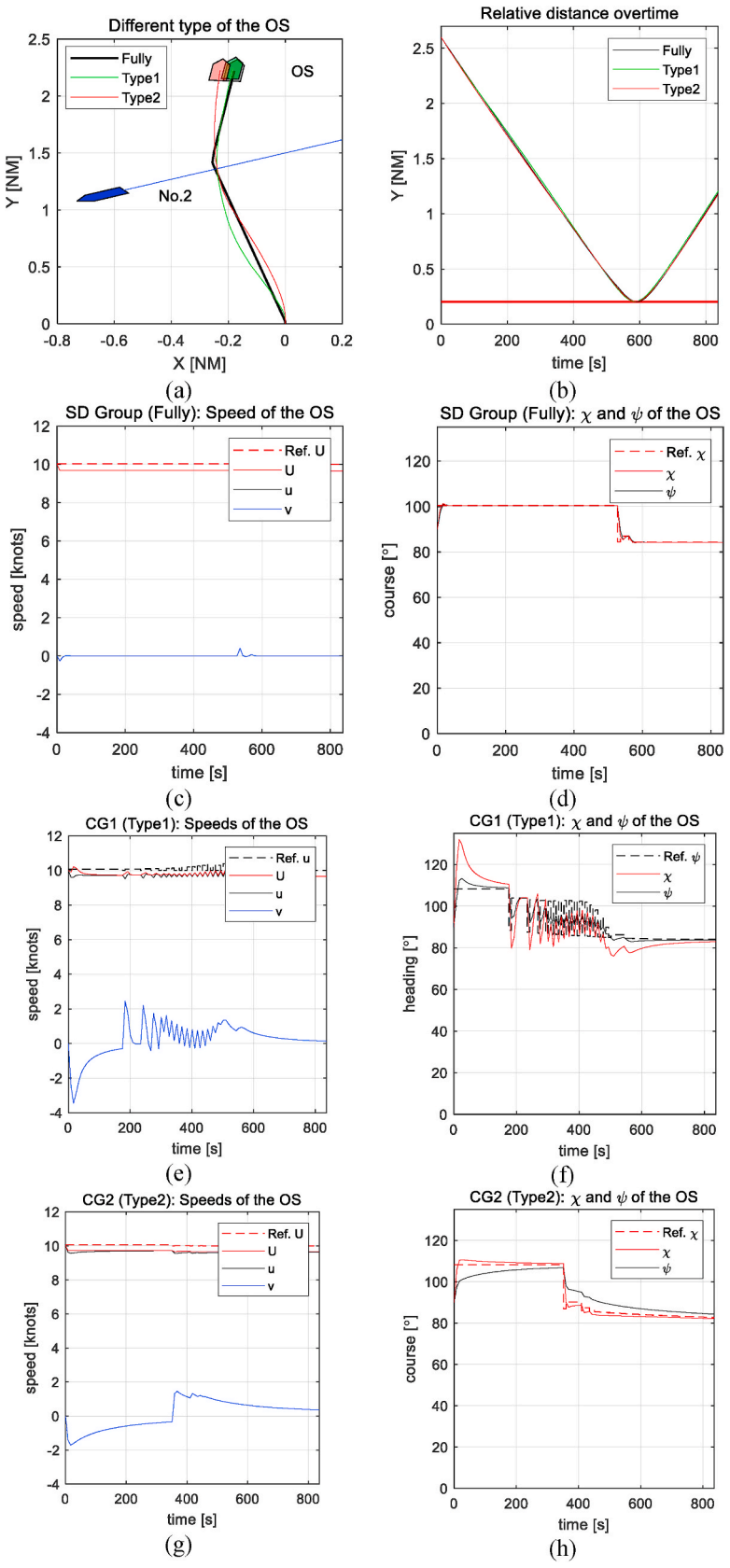


Fig. 10. The HMI-CAS in Auto mode under different ship dynamics assumptions.

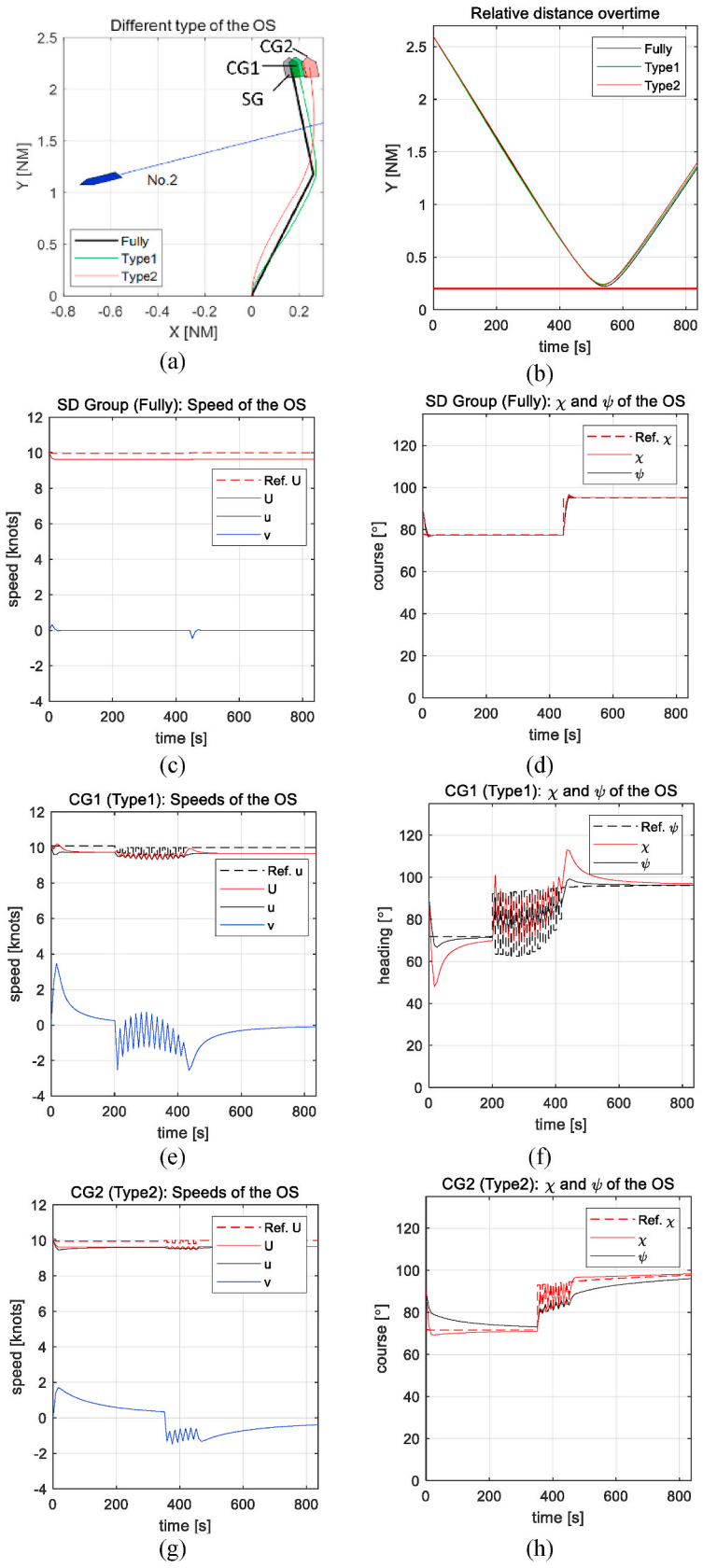


Fig. 11. The HMI-CAS in Manual mode under different ship dynamics assumptions.

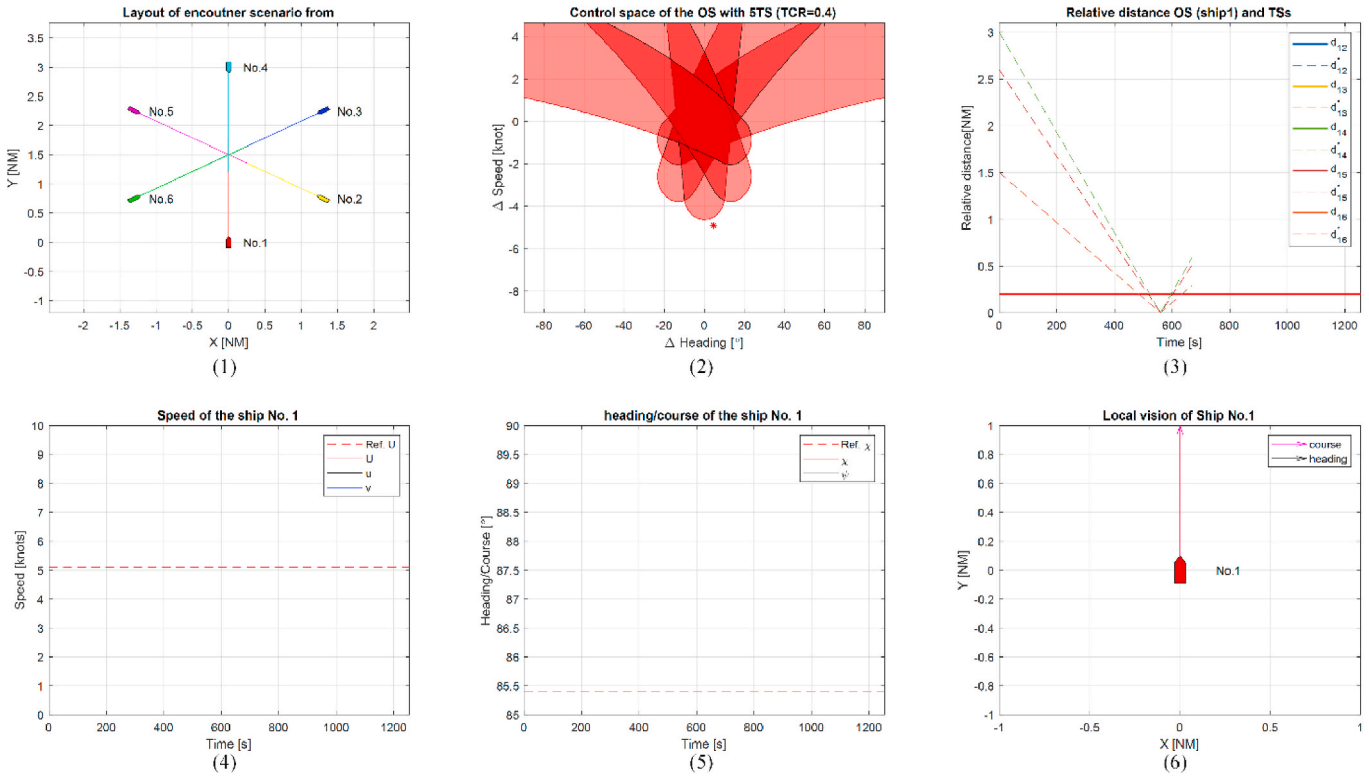


Fig. 12. Multiple-encounter Scenario from the vision of the NO. 1 ship (or OS) at $t = 0$ [s].

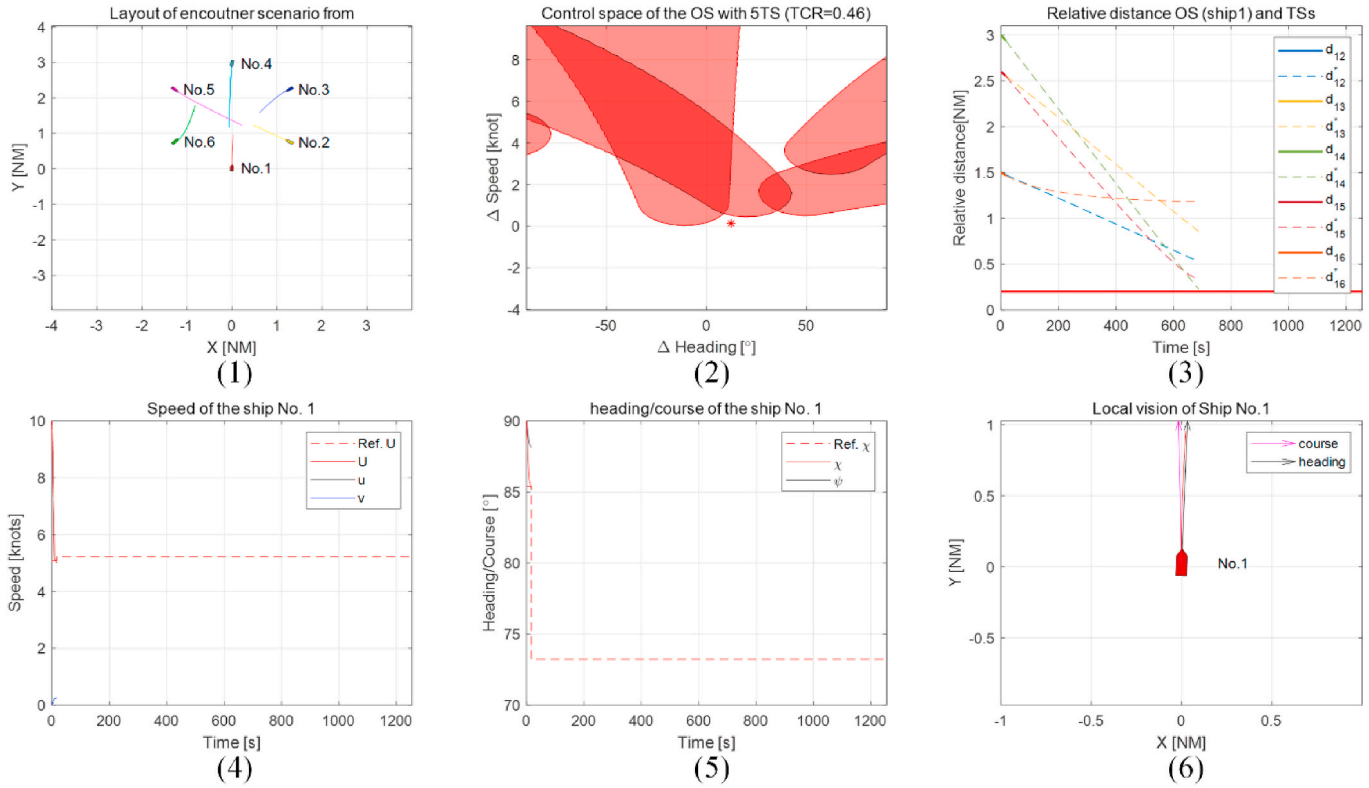


Fig. 13. Multiple-encounter Scenario from the vision of the NO. 1 ship (or OS) at $t = 16$ [s].

new solution is found. The new solution basically requires the ship to keep its resultant speed, but to increase its starboard turning by 13°. That means the course of the ship changes from 86° to 73°.

Fig. 14 shows the vision from the "NO. 1" ship at the end of the simulator. From the relative distance over time (panel (3)), it is observed that all the relative distances are larger than the safety distance. It means

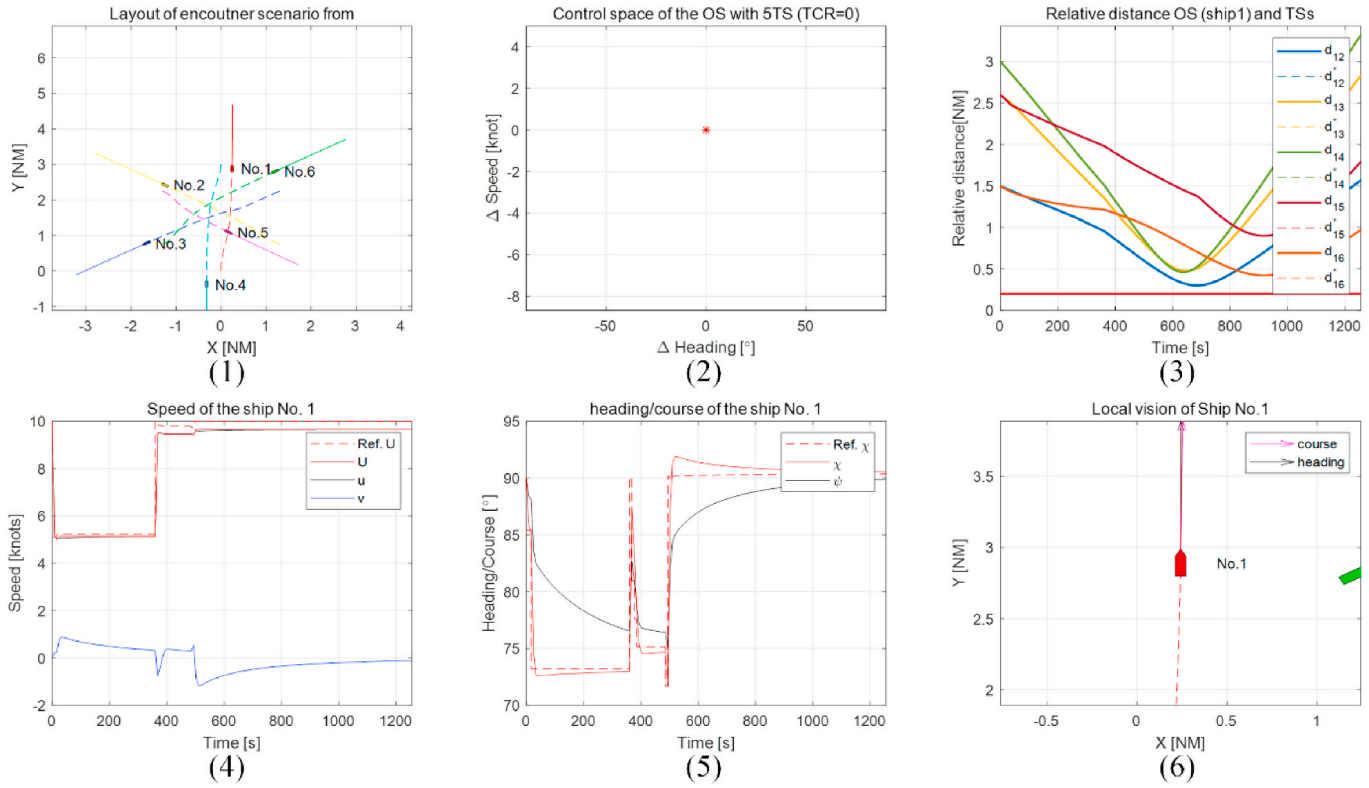


Fig. 14. Multiple-encounter Scenario from the vision of the NO. 1 ship (or OS) at the end.

that the ship avoids collisions safely. Panel (4) and (5) show the recorded speeds and courses over time.

5. Discussion

In this section, the performance and the feature of the HMI-CAS are discussed, followed by the limitations found during simulations and the relevant solutions.

5.1. A portable connecting human knowledge and automation via HMI-CAS

As shown in Section 5.2.1, the automatic system might offer a solution that is violating rules or not encouraged by navigators. To prevent this kind of behavior, human operators are in charge of monitoring the system.

The traditional collision avoidance methods do not offer the portable that facilitates human operators to join the collision avoidance process. Firstly, some solutions of CAS are unfriendly for human operators to read and understand, e.g. a series of forces at different moments. Secondly, some solutions offer one collision-free solution with no information for human operators to modify and judge the solution, e.g. one optimal trajectory. Subsequently, if the operators do not satisfy the solution suggested by machines, they have to analyze the encounter situation again, find a new solution, and validate their new solution by their experience.

The proposed HMI-CAS, in contrast, offers the portable that allows the cooperation between human operators and the automatic systems. The human and the machine can both get benefit from the cooperation.

The HMI-CAS visualizes the solution space marking dangerous solutions that helps human operators in the following three aspects. Firstly, the human operators, then, can judge the safety of the ship by checking the location of the existing velocity in the space. Secondly, since the optimal solution by the system is also displayed in the solution space, the operators can easily judge whether the solution is violating regulations or not, see Section 5.2.2. Thirdly, the operators can validate their new solution via the solution space if they do not agree with the suggested solution. In brief, the human operators are supported by the machine to read the solution, to understand, and to intervene in the automatic system.

The human operators, on the other hand, also support the automatic system. Firstly, human operators help the automatic system to be rule-compliant. To date, the rule-compliant automatic system is still challenging in various automation. Specifically, the interpretation of regulations depends on navigators' experience and good seamanship that differ from one another. Using the proposed HMI-CAS, the operators can use their own experience to judge the machines' solution. If the solution is violating rules or the good seamanship, the operators, then, can modify the solution by inputting a new solution. Secondly, operators help the automation to find a solution when there is no collision-free solution by using Auto mode. The Auto mode is based on the approximation of the feasible-solution space, see Fig. 8. When we use a half-plane to represent the dangerous space, some collision-free solutions are inevitably excluded. Thus, in an extreme situation, it is possible that there are no solutions in Auto mode, while the human operators can easily identify a collision-free solution out of UO sets.

In brief, the development of the automatic system in MASS is not completely excluding the human from the system, but incorporating human's intelligence to contribute to a safer and smarter system.

5.2. Potential of the proposed HMI-CAS

There are some potentials of the proposed HMI-CAS that also can be expected.

Firstly, it is possible to accept different cost functions. In Equation (19), the Auto mode uses a quadratic cost function that minimizes the changes in controls. It is also possible to incorporate a cost function minimizing the consumption of fuel oil.

Secondly, it also helps manned ships to cooperate with fully unmanned ships proposed in (Chen et al., 2018). For manned ships, how to

cooperate with unmanned ships or vice versa is an unsolved problem. The proposed system offers a possible for the manned ship to cooperate with the unmanned ship. Specifically, the HMI-CAS helps human operators to communicate with unmanned ships and visualizes the solutions leading to a collision with unmanned ships.

Thirdly, the proposed HMI-CAS enables to suggest the time for applying the last maneuvers. In literature (Huang and Van Gelder, 2020), a time-varying collision risk (TCR) measure is proposed, which is measured as the room-for-maneuver. The TCR level for the OS is presented in Fig. 9. TCR reaches 1 means there are no collision-free solutions found by the machine. Therefore, the operators can postpone the suggested solution by the machine until the TCR reaches 1. After the TCR reaches 1, the cooperation between ships is necessary to prevent accidents.

5.3. Eliminating oscillation in the HMI-CAS

According to the simulation results, the underactuated ship may have oscillations during collision avoidance using HMI-CAS, see Figs. 9 and 10. Thus, we conclude that the oscillation becomes larger when the ship is underactuated. Compared with CG2, it is observed that the oscillation is much severe when the ship cannot control the sway speed (CG1). The oscillation behavior might be incompliant with the COLREGs and maritime practice. Thus, it requires further studies to restrain the frequency of the oscillation.

In (Van Den Berg et al., 2008), the oscillation behavior of the vehicle using the velocity obstacle algorithm has been reported, where the oscillation is generated due to the incoordination between ships. Specifically, two vehicles make the decision together and both assume that the other ship continues the existing speed.

However, in this manuscript, the cause of oscillation is different since the ships are coordinated and updated their behavior one-be-one. The oscillation behavior appearing in this manuscript is mainly caused by the errors due to the successive linearization in Equation (6). The errors due to linearization would cause the boundary of the UO set slightly different from it is presented in solution space. Therefore, when the OS detects the reference velocity is just out of UO sets, it immediately changes its desired velocity to the reference velocity, i.e., $\mathbf{u}^* = \mathbf{r}$. However, when the ship steers to the desired velocity, the linearized system is also updated. The new UO set based on the updated linearized system might indicate the desired velocity is unsafe. Thus, the ship steers back to the original collision-free velocity.

To reduce the oscillations of under-actuated ships, it is recommended to accept the setting of CG2 that tracks the course and resultant speed instead of the heading and surge speed. Moreover, in Section 4.4, more measures are introduced to reduce oscillations. For instance, the UO set should be enlarged, e.g., a larger repulsive term $\widehat{w}$ is considered; the ship is designed not to change its desired input frequently, e.g., Rule 2 and Rule 3 are introduced. These measures can reduce the oscillation but cannot prevent it. Another approach is including reachability analysis in the construction of UO sets. Specifically, the upper bounds of the errors are estimated by reachability analysis and the UO sets are enlarged accordingly. This idea would be discussed and considered in future studies.

5.4. Control authority transition between human mode to machine mode

As we demonstrated in Section 5.2, the control authority is switched to human operators when the machine detects inputs from human operators. Specifically, when operators observe the possible violation of rules, the operators can choose a new collision-free solution out of UO set by input devices, e.g. touch screen, etc.; when operators do not input a new decision, the Auto Mode would execute the solution automatically, i.e. automatic collision avoidance.

The existing setting of control transition can be used in MASS I-III, while the human operators (onboard or in the remote-control centre) should monitor the HMI-CAS all time, which is similar to the L2 level of driving automation (SAE International, 2016). To achieve a higher level of driving automation, i.e. conditional automation (L3), high automation (L4), etc., the studies on the transition of control authority should be added to the proposed HMI-CAS. These studies include: the timing of control transition; the way of early alert and decision supports that help operators to build situation awareness as soon as possible; the mixture of control during control transition, etc. Some results from self-driving cars can offer several clues, while rigorous tests for the maritime environment are still needed since the working environment, the background of drivers, certificate of skill, etc. Are so different.

5.5. Conclusions and future research

In the transition to autonomous eras, the demand for human-machine interaction (HMI) is increasing. However, few existing Collision Avoidance Systems (CASs) can support the cooperation between human operators and the automatic system during collision avoidance.

In this article, an CAS that is interpretable and interactive for human operators is proposed, which is called Human-Machine Interaction oriented Collision Avoidance System (HMI-CAS). The HMI-CAS presents its collision-free decision and the solution space to human operators, so that human operators can read the intentions of the machines, understand collision-free decision offered by the machine, and intervene in the decision-made making processes of the machines during collision avoidance. The proposed system aims at helping human operators and machine to share their intelligence in solving collision avoidance problem, i.e. hybrid intelligence combining the complementary strengths of human operators and machines.

By comparing existing collision avoidance algorithms, a family of velocity obstacle algorithms has been found, which firstly visualizes the solution space and then finds an optimal solution. Specifically, to consider the under-actuated dynamics of the ship, a generalized velocity obstacle (GVO) algorithm has been applied to the proposed HMI-CAS system. To handle the under-actuated dynamics, two modes of control are proposed: Control Group 1 (CG1), the ship only tracks heading and surge speed; CG2, the ship tracks course and resultant speed.

To test and to demonstrate the performance of the HMI-CAS on board, several simulations are introduced, where the ship is assumed to be fully actuated (Standard Group, SG) and under-actuated (CG1 and CG2). The results reveal that all these settings can support the ship to avoid collision automatically and support HMIs. Moreover, since the solution space is visualized, it is convenient for human operators to detect rule-violation behavior and to intervene in the automation during collision avoidance. By comparing the collision avoidance process in SG, CG1 and CG2, we also observed that the setting from CG1 would easily result in severe oscillations during collision avoidance, which would be dangerous for the ship. Thus, we recommend accepting CG2 settings for the HMI-CAS when the ship is under-actuated.

With the proposed HMI-CAS, the machine visualizes the solution space to human operators and makes the automation process transparent for human users, which helps them read, understand, and intervene in the system. On the other hand, human operators can help the automatic system be rule-compliant. The potentials of the HMI-CAS include: accept different cost functions; helps manned ships to cooperate with fully unmanned ships; suggest the time for applying the last maneuver.

Further research considers the following directions: (1) testing the HMI-CAS with different model ships in simulator environment and physical experiments; (2) introducing the studies on reachability analysis for reducing the oscillation; (3) investigating the characteristics of human-machine interactions in the maritime environment; (4) considering the dynamics of rudder servo systems and the irregular shapes of the ships; (5) considering uncertainty of other ships' behavior by identifying ship patterns (e.g., the technique presented in (Huang et al., 2020b).

CRediT authorship contribution statement

Yamin Huang: Conceptualization, Methodology, Data curation, Writing - original draft, Visualization, Writing - review & editing. **Linying Chen:** Conceptualization, Methodology, Writing - review & editing, Investigation. **Rudy R. Negenborn:** Funding acquisition, Resources. **P.H.A.J.M. van Gelder:** Resources, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This research is supported by the Fundamental Research Funds for the Central Universities in Wuhan University of Technology (WUT: 203144001, 203112003) and the Researchlab Autonomous Shipping in Delft University of Technology.

Abbreviations

APF	Artificial Potential Field
CASs	Collision Avoidance Systems
COLREGs	Convention on the International Regulations for Preventing Collisions at Sea
ConfP	Conflict Positions
DD	Decision Disc
DW	Dynamic Window
FMM	Fast Marching Method
GNC	Guidance, Navigation, and Control
GVO	Generalized VO
HMI	Human-Machine Interaction
HMI-CAS	HMI oriented Collision Avoidance Systems
LCM	Limited Cycle Method
MASS	Maritime Autonomous Surface Ships
MPC-CA	Model Predictive Control based
CG	Control Group Collision Avoidance
OS	Own Ship
PD	Proportional-derivative
POA	Projected Obstacle Area
RK	Runge-Kutta
SG	Standard Group
sUO	sub-UO set
TCR	Time-varying Collision Risk
TS	Target Ship
VC	Visual Cone
VO	Velocity obstacle

Notations

a	Wight in cost function J_{UO}
$g(x)$	observe function
$G(t)$	Response function to the $\Delta\mathbf{u}$
J_{UO}	Quadratic cost function
$K_{d,p}$	Feedback gains
N	Torque
$\boldsymbol{O}$	the origin of the control space
$\mathbf{P}$	Position of ship
r	Angular velocity/yaw
$\mathbf{r}$	Reference velocity to the system
sUO(t)	Sub-UO set at time t
T	Thrust
u	Surge Speed
$\breve{u}$	Desired surge speed
$\mathbf{u}$	Desired velocity vector
U	Desired resultant speed
U	The control space of the ship
UO	A set in control space U lead to collision
v	Sway Speed
$\mathbf{v}$	Velocity vector consists of u, v, and r
V	2-by-6 matrix
$\mathbf{w}$	Closest point on the boundary to $\mathbf{u}^0$

$\widehat{\mathrm{w}}$	A repulsive term adding to **W**
x	Position in x-axis
$\mathbf{x}$	State vector of the ship
y	Position in y-axis
$\boldsymbol{\tau}$	Input vector consists of thrust and torque
χ	Desired course
ψ	Heading of the ship
$\breve{\psi}$	Desired heading
{b}	Ship body frame
$\{n\}$	Inertial frame

Matrices

$\mathbf{0}$	Zero matrix
A	Linearized system matric w.r.t. State $\mathbf{x}$ at $\mathbf{x}^0$
B	Linearized input matric w.r.t. input $\mathbf{u}$ at $\mathbf{u}^0$
$\mathbf{B}$	Input matrix
C	Output matrix
$\mathrm{C(v)}$	Coriolis-centripetal matrix
$\mathbf{D(v)}$	Damping matrix
$\mathbf{I}$	Identity matrix
$\mathbf{M}$	Mass matrix consists of rigid-body mass and added mass

Operators, Superscripts, and Subscripts

$\mathrm{CH}(\square)$	An operator returning convex hull
$\overline{\square}$	Complement of the value set
$\mathrm{R}(\square)$	An operators returning rotated vector
$\dot{\square}$	Derivative of the value
$\partial\square$	An operator returning the boundary of a set
$\tilde{\square}$	Estimation of the value
$\Delta\square$	Difference between$\square$ and $\mathbf{u}^0$
$\square_i$	Ship i related value
$\square^0$	Initial state
$\square_f$	Ship j related value
$\square^*$	Optimal value
$\square_{\min}$	Minimal value
$\square^{bound}$	Boundary of the value
$\square_{\max}$	Minimal value
$\square^{fea}$	Feasible range of the value
$\square_{m\times n}$	A m-by-n matrix
$\square^{\mathrm{human}}$	Human determined value
$\square_{u,v}$	A value in surge/heading direction

References

Abdelaal, M., Franzle, M., Hahn, A., 2018. Nonlinear Model Predictive Control for trajectory tracking and collision avoidance of underactuated vessels with disturbances. Ocean. Eng. 160, 168–180.

Alonso-Mora, J., Beardsley, P., Siegwart, R., 2018. Cooperative collision avoidance for nonholonomic robots. IEEE Trans. Robot. 34, 404–420.

Benjamin, M.R., Leonard, J.J., Curcio, J.A., Newman, P.M., 2006. A method for protocol-based collision avoidance between autonomous marine surface craft. J. Field Robot. 23, 333–346.

Campbell, S., Naeem, W., Irwin, G.W., 2012. A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres. Annu. Rev. Contr. 36, 267–283.

Chen, L., Hopman, H., Negenborn, R.R., 2018. Distributed model predictive control for vessel train formations of cooperative multi-vessel systems. Transport. Res. C Emerg. Technol. 92, 101–118.

Degre, T., Lefevre, X., 1981. A collision avoidance system. J. Navig. 34, 294–302.

Fang, M.-C., Tsai, K.-Y., Fang, C.-C., 2017. A simplified simulation model of ship navigation for safety and collision avoidance in heavy traffic areas. J. Navig. 71, 837–860.

Ferranti, L., Negenborn, R.R., Keviczky, T., Alonso-Mora, J., 2018. Coordination of Multiple Vessels via Distributed Nonlinear Model Predictive Control. *17th European Control Conference (ECC 2018)*. Limassol. Cyprus.

Fossen, T.I., 2002. Marine Control Systems: Guidance, Navigation, and Control of Ships, Rigs and Underwater Vehicles, Trondheim, Norway. Marine Cybernetics.

Goerlandt, F., Montewka, J., Kuzmin, V., Kujala, P., 2015. A risk-informed ship collision alert system: framework and application. Saf. Sci. 77, 182–204.

Graziano, A., Teixeira, A.P., Guedes Soares, C., 2016. Classification of human errors in grounding and collision accidents using the TRACEr taxonomy. Saf. Sci. 86, 245–257.

Huang, Y., Chen, L., Chen, P., Negenborn, R.R., Van Gelder, P.H.A.J.M., 2020a. Ship collision avoidance methods: state-of-the-art. Saf. Sci. 121, 451–473.

Huang, Y., Van Gelder, P.H.A.J.M., 2020. Time-varying risk measurement for ship collision prevention. Risk Anal. 40, 24–42.

Huang, Y.M., Chen, L.Y., Van Gelder, P.H.A.J.M., 2019. Generalized velocity obstacle algorithm for preventing ship collisions at sea. Ocean. Eng. 173, 142–156.

Huang, Y.M., Van Gelder, P.H.A.J.M., Wen, Y.Q., 2018. Velocity obstacle algorithms for collision prevention at sea. Ocean. Eng. 151, 308–321.

Huang, L., Wen, Y., Guo, W., Zhu, X., Zhou, C., Zhang, F., Zhu, M., 2020b. Mobility pattern analysis of ship trajectories based on semantic transformation and topic model. Ocean. Eng. 201, 107092.

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. Transport. Syst. 17, 3407–3422.

Kim, H., Kim, D., Shin, J.-U., Kim, H., Myung, H., 2014. Angular rate-constrained path planning algorithm for unmanned surface vehicles. Ocean. Eng. 84, 37–44.

Koopman, P., Wagner, M., 2017. Autonomous vehicle safety: an interdisciplinary challenge. IEEE Intelligent Transportation Systems Magazine 9, 90–96.

Kuwata, Y., Wolf, M.T., Zarzhitsky, D., Huntsberger, T.L., 2014. Safe maritime autonomous navigation with COLREGS, using velocity obstacles. IEEE J. Ocean. Eng. 39, 110–119.
Larson, J., Bruch, M., Ebken, J., 2006. Autonomous navigation and obstacle avoidance for unmanned surface vehicles. In: Gerhart, G.R., Shoemaker, C.M., Gage, D.W. (Eds.), SPIE 6230, Unmanned Systems Technology VIII, 623007. Orlando, Florida, United States. 623007-1-12.
Lenart, A.S., 1983. Collision threat parameters for a new radar display and plot technique. J. Navig. 36, 404–410.
Li, S., Liu, J., Cao, X., Zhang, Y., 2018. A novel method for solving collision avoidance problem in multiple ships encounter situations. In: Cerulli, R., Raiconi, A., Voß, S. (Eds.), *International Conference on Computational Logistics (ICCL 2018)*. Vietri Sul Mare (Salerno). Italy Springer. Cham.
Liu, C., Mao, Q., Chu, X., Xie, S., 2019. An Improved A-Star Algorithm Considering Water Current, Traffic Separation and Berthing for Vessel Path Planning, vol. 9. Applied Sciences.
Liu, Y., Bucknall, R., 2016. The angle guidance path planning algorithms for unmanned surface vehicle formations by using the fast marching method. Appl. Ocean Res. 59, 327–344.
Liu, Y.C., Liu, W.W., Song, R., Bucknall, R., 2017. Predictive navigation of unmanned surface vehicles in a dynamic maritime environment when using the fast marching method. Int. J. Adapt. Contr. Signal Process. 31, 464–488.
Liu, Z.X., Zhang, Y.M., Yu, X., Yuan, C., 2016. Unmanned surface vehicles.: an overview of developments and challenges. Annu. Rev. Contr. 41, 71–93.
Loe, Ø.A.G., 2008. Collision Avoidance for Unmanned Surface Vehicles. Master of Science. Norwegian University of Science and Technology.
Lyu, H., Yin, Y., 2018. COLREGS-constrained real-time path planning for autonomous ships using modified artificial potential fields. J. Navig. 72, 588–608.
Lyu, H.G., Yin, Y., 2017. Ship's trajectory planning for collision avoidance at sea based on modified artificial potential Field. 2017 2nd International Conference on Robotics and Automation Engineering (Icrae) 351–357.
Mahini, F., Diwilliams, L., Burke, K., Ashrafiuon, H., 2013. An experimental setup for autonomous operation of surface vessels in rough seas. Robotica 31, 703–715.
Naeem, W., Irwin, G.W., Yang, A.L., 2012. COLREGs-based collision avoidance strategies for unmanned surface vehicles. Mechatronics 22, 669–678.
Ożoga, B., Montewka, J., 2018. Towards a decision support system for maritime navigation on heavily trafficked basins. Ocean. Eng. 159, 88–97.
Pedersen, E., Inoue, K., Tsugane, M., 2003. Simulator studies on a collision avoidance display that facilitates efficient and precise assessment of evasive manoeuvres in congested waterways. J. Navig. 56, 411–427.
Perera, L.P., Carvalho, J.P., Guedes Soares, C., 2012. Intelligent ocean navigation and fuzzy-bayesian decision/action formulation. IEEE J. Ocean. Eng. 37, 204–219.
SAE International, 2016. Levels of driving automation are defined in new SAE International standard. SAE International, J3016.
Schiaretti, M., Chen, L., Negenborn, R.R., 2017. Survey on autonomous surface vessels: Part II - categorization of 60 prototypes and future applications. Comput. Logist. 234–252. https://doi.org/10.1007/978-3-319-68496-3_16.
Serigstad, E., 2017. Hybrid Collision Avoidance for Autonomous Surface Vessels. NTNU.
Shah, B.C., Švec, P., Bertaska, I.R., Sinisterra, A.J., Klinger, W., Von Ellenrieder, K., dhanak, M., gupta, S.K., 2015. Resolution-adaptive risk-aware trajectory planning for surface vehicles operating in congested civilian traffic. Aut. Robots 40, 1139–1163.
Soltan, R.A., Ashrafiuon, H., Muske, K.R., 2009. Trajectory Real-Time Obstacle Avoidance for Underactuated Unmanned Surface Vessels, pp. 1059–1067.
Soltan, R.A., Ashrafiuon, H., Muske, K.R., 2010. ODE-based obstacle avoidance and trajectory planning for unmanned surface vessels. Robotica 29, 691–703.
Švec, P., Thakur, A., Raboin, E., Shah, B.C., Gupta, S.K., 2013. Target following with motion prediction for unmanned surface vehicle operating in cluttered environments. Aut. Robots 36, 383–405.
Szlapczynski, R., Krata, P., 2018. Determining and visualizing safe motion parameters of a ship navigating in severe weather conditions. Ocean. Eng. 158, 263–274.
Tam, C., Bucknall, R., Greig, A., 2009. Review of collision avoidance and path planning methods for ships in close range encounters. J. Navig. 62, 455–476.
Van Den Berg, J., Lin, M., Manocha, D., 2008. Reciprocal Velocity Obstacles for Real-Time Multi-Agent Navigation. *2008* IEEE International Conference on Robotics and Automation (ICRA). IEEE, Pasadena, CA, USA.
Wiig, M.S., Pettersen, K.Y., Krogstad, T.R., 2017a. A Reactive Collision Avoidance Algorithm for Vehicles with Underactuated Dynamics.
Wiig, M.S., Pettersen, K.Y., Savkin, A.V., 2017b. A Reactive Collision Avoidance Algorithm for Nonholonomic Vehicles. *2017* Ieee Conference on Control Technology and Applications (Ccta 2017), pp. 1776–1783.
Wu, B., Cheng, T., Yip, T.L., Wang, Y., 2020. Fuzzy logic based dynamic decision-making system for intelligent navigation strategy within inland traffic separation schemes. Ocean. Eng. 197, 106909.
Xue, Y., Clelland, D., Lee, B.S., Han, D., 2011. Automatic simulation of ship navigation. Ocean. Eng. 38, 2290–2305.
Xue, Y., Lee, B.S., Han, D., 2009. Automatic collision avoidance of ships. Proc. IME M J. Eng. Marit. Environ. 223, 33–46.
Zhang, J.F., Zhang, D., Yan, X.P., Haugen, S., Soares, C.G., 2015. A distributed anti-collision decision support formulation in multi-ship encounter situations under COLREGs. Ocean. Eng. 105, 336–348.
Zhao-Lin, W., 1988. Analysis of radar PAD information and a suggestion to reshape the PAD. J. Navig. 41, 124–129.
Zheng, H., 2016. *Coordination of Waterborne AGVs*. PhD. Delft University of Technology.
Zhuang, J.Y., Zhang, L., Zhao, S.Q., Cao, J., Wang, B., Sun, H.B., 2016. Radar-based collision avoidance for unmanned surface vehicles. China Ocean Eng. 30, 867–883.