The Journal of Navigation (2022), **75**:5 1243–1267
doi:10.1017/S0373463322000315

RESEARCH ARTICLE

Agile collision avoidance for unmanned surface vehicles based on collision shielded model prediction control algorithm

Yihan Tao, and Jialu Du*

School of Marine Electrical Engineering, Dalian Maritime University, Dalian 116026, China.
*Corresponding author. E-mail: dujl@dlmu.edu.cn

Received: 20 July 2021; **Accepted:** 15 May 2022; **First published online:** 18 July 2022

Keywords: Unmanned surface vehicle; autonomous; ship collision

Abstract

Collision avoidance (COLAV) is a prerequisite for the navigation safety of unmanned surface vehicles (USVs). Since USVs have to avoid obstacles clearly and timely, i.e. the COLAV should be agile, the COLAV algorithm should have low computation complexity and make efficient COLAV decisions. However, balancing between the computation complexity and the COLAV decision optimality is still intractable at present. This paper innovatively proposes a COLAV algorithm for USVs by combining the velocity obstacle method with the predictive model method, named the collision shielded model predictive control (CS-MPC) algorithm, such that the agility of USVs COLAV is improved. The runtime of the proposed COLAV algorithm is shortened by shielding the dangerous parts of the search space of the COLAV decisions, and the COLAV decision is efficient with the aid of the accurately predicted motion trajectory by the motion mathematical model of USVs. As such, the USV can safely navigate in complex water areas where multiple vessels and obstacles exist. A series of simulations on a yacht in different kinds of encounter situations were carried out to verify the effectiveness and the agility of the proposed CS-MPC COLAV algorithm.

1. Introduction

With the recovery of international trade and the shipping industry, the sailing environment of ships is becoming increasingly more complex, and the incidence of ship collision accidents is always high in global waters. According to the International Maritime Organisation (IMO) survey, human factors are responsible for more than 75% of all maritime accidents worldwide (Levander, 2017). The safety of maritime traffic is still grim because of the human factor. Unmanned surface vehicles (USVs) are one of the solutions to reduce the number of ship collision accidents caused by the human factor. At the same time, with the development of modern science and technology, the world is about to enter the era of ‘Industry 4.0.’ Many fast-growing technologies for maritime vessels, especially those in guidance, navigation, and control systems (GNC), promote the evolution of autonomous navigation through ship automation. Therefore, USVs are currently being applied more widely for their safer navigation (Kongsberg Maritime, 2021).

Above all, the development of USVs highly depends on navigation safety. Furthermore, one of the most important advantages of USVs is the autonomous collision avoidance (COLAV) capability. However, it is very challenging for USVs to avoid collisions, especially in close-quarters situations. Most USVs are featured by the underactuations and nonlinearities, and they have to avoid static and dynamic obstacles clearly and timely to comply with the rules of the International Regulations for Preventing

https://doi.org/10.1017/S0373463322000315 Published online by Cambridge University Press

Collisions at Sea (COLREGs). Therefore, the COLAV must be agile, and the COLAV algorithms should have low computation complexity and make optimal COLAV decisions.

To date, there are a large number of COLAV algorithms for USVs (Liu et al., 2016). Some mature algorithms are quite widely used, such as the artificial potential field (APF) method, the velocity obstacle method (VO), the dynamic window approach (DWA), the constant avoidance angle method (CAA) and the model predictive control (MPC) algorithm. Furthermore, the logic-based decision (James, 1986; Zheng, 2000; Perera et al., 2011; Kreutzmann et al., 2013), the set-based decision (Benjamin et al., 2006; Moe and Pettersen, 2017; Kufoalor et al., 2020) and other theories also provide methods and ideas. In addition, some path planning algorithms can also be used to solve the COLAV problem by frequently replanning in a short time, i.e. the COLREGs rapidly exploring random tree (COLREGs-RRT) algorithm (Chiang and Tapia, 2018; Enevoldsen et al., 2021). However, these planning algorithms are not discussed in detail because they are not used as typical COLAV algorithms.

The APF method was first proposed by Khatib (1986). Lyu and Yin (2018) modified this algorithm and applied it to solve the COLAV problem of USVs while ensuring the COLAV decision solutions would not get stuck in the local optimal ones. However, the APF method cannot predict the future situations. When the position of obstacles and the traffic situation are complex, the USVs sometimes have to sail through a very complicated process to converge to planned paths. The long sampling time also reduces the agility of USVs to avoid collision.

The VO method, also known as the velocity cone method, was first proposed by Fiorini and Shiller (1998), and then extended into many branches, such as the nonlinear velocity obstacle (Martinezgomez and Fraichard, 2009) and the reciprocal velocity obstacle (Snape et al., 2011). Kufoalor et al. (2018) proposed a dynamic reciprocal velocity obstacle method that incorporates the interactive behaviour of agents and is proactive in dealing with the uncertainty of the future behaviour of obstacles. The advantages of this method lie in that it is intuitive, simple to calculate and easy to apply COLREGs (Zhuang et al., 2011; Kuwata et al., 2014). Stenersen (2015) applied the VO algorithm into the ROS system (Open Robotics, 2021) and built a simulation experimental environment for COLAV of USVs. Huang et al. (2019) introduced the USV dynamics to the VO algorithm by modifying the generalised VO (GVO) algorithm and applying the GVO algorithm to a human-machine interaction collision avoidance system (Huang et al., 2020). However, the headings of USVs may oscillate, the trajectories may not be optimal and the destinations are sometimes blocked by the velocity cones.

The dynamic window approach is one of the most widely used algorithms in the field of intelligent guidance at present. It was first proposed by Fox et al. (1997), and has been widely extended and applied (Brock and Khatib, 1999; Kiss and Tevesz, 2012). Eriksen et al. (2018) replaced the robot motion mathematical model in the dynamic window method with a three-dimension-of-freedom (3DOF) ship motion model, enabling the algorithm to consider the underactuated and nonlinear features of USVs. However, predicting the trajectories numerically will increase the computation complexity.

The CAA method was first proposed by Savkin and Wang (2013). Wiig et al. (2019) introduced the hydrodynamics of the USVs into the CAA algorithm, enabling USVs to avoid obstacles only by changing courses. It is suitable for most maritime surface vessels. However, the CAA algorithm is only applicable to the encounter situations where the target ships are sparsely distributed, and the static obstacles in the actual environment are not taken into account at present.

The model predictive control (MPC) algorithm was first proposed by Richalet et al. (1978). Hagen et al. (2018) and Kufoalor et al. (2019) made the MPC algorithm well applied to the USV COLAV by modifying the objective function. However, to reduce the computation complexity, the search space is sampled sparsely, which may increase the distance between the approached results and the optimal solutions.

Table 1 shows the comparisons of the approaches mentioned above. The VPF and VO methods have low computational complexity and short runtime. However, without the accurate prediction of the USV's trajectory, these algorithms cannot make protective manoeuvres, which means never choosing the dangerous manoeuvres in advance. As a result, there are some difficulties in dealing with local problems in specific environments, such as chattering, being stuck or blocking goals. Although the CAA algorithm

https://doi.org/10.1017/S0373463322000315 Published online by Cambridge University Press

can overcome some local problems, it cannot be applied in a complex environment complying with the COLREGs rules at present. Therefore, these kinds of algorithms should still be improved in the aspects of optimality. The above local problems can be well solved by MPC-based algorithms. However, for the underactuated USVs, the trajectory prediction by the numerical method increases the computation complexity.

In conclusion, although the USV COLAV algorithms proposed in recent years are able to deal with the COLAV problem with different advantages, balancing the trade-off between the computational performance and the solution optimality appears intractable. Therefore, the agility of the COLAV algorithm still needs to be improved.

To improve the agility of COLAV, a modified model predictive control (MPC) algorithm is proposed for USVs, named collision shielded MPC (CS-MPC). The innovations of the proposed method can be concluded as the following two aspects:

(1) the computation complexity is reduced by shielding the dangerous velocities in the search space before the trajectory prediction, based on the velocity obstacle method;
(2) better COLAV decisions are obtained by introducing a nonlinear ship motion model and a modified objective function.

Compared with other COLAV algorithms, the CS-MPC has a larger scale of optional COLAV manoeuvres by optimising the search space, which helps to find better manoeuvres to steer around obstacles blocking the goal. Furthermore, because of the reduced computing complexity, it has a longer predicted time, which will help reduce the chance of being sucked. By defining the COLREGs penalty in the objective function, the COLAV manoeuvres comply with the COLREGs and the chattering problem is solved.

Combined with velocity controllers, the CS-MPC algorithm can also guide USVs to destinations. It can be used not only for COLAV but also for trajectory tracking. The safe autonomous navigation in complicated waters can be achieved.

The rest of the paper is structured as follows. Section 2 describes the COLAV problem to be studied. Section 3 proposes the CS-MPC algorithm. To verify the agility improvement of the CS-MPC algorithm, the results of simulation experiments are shown in Section 4. Section 5 discusses the simulation results and possible improvements to the CS-MPC algorithm. Section 6 concludes the whole paper and interprets the implications of the CS-MPC algorithm.

2. Collision avoidance problem

For its rapid-reaction capability, the collision avoidance (also called the local collision avoidance) algorithm is one of the core algorithms of the USV GNC system, ensuring navigation safety. The objective of the COLAV algorithm is to select the optimal desired velocities. It should consider the dynamic constraints of the vessel, such as underactuations and nonlinearities of USV motion and the limitations of controllers.

2.1. Architecture of USVs GNC systems

The local COLAV, mid-level decision-making and global path-planning subsystems are hierarchical algorithms in a USV's guidance system (Larson et al., 2006; Eriksen, 2019; Yang, 2019; Loe, 2008). The local COLAV subsystem receives the planed trajectories which are produced by upper algorithms and the information of the USV and environment from the navigation system, which consists of information fusion, environmental perception, state estimation and situation awareness technologies. The architecture of the USV GNC system is shown in Figure 1.

This paper focuses on the COLAV algorithm, so the following assumptions are made.

https://doi.org/10.1017/S0373463322000315 Published online by Cambridge University Press

Table 1. *Comparisons of COLAV algorithms.*

Method		APF	VO		CAA	DWA	MPC	
Authors		Lyu and Yin	Stenersen	Huang	M.S. Wiig	Eriksen	Kufoalor and Hagen	Tao and Du
Years		2018	2015	2020	2019	2018	2019	This paper
USV motion prediction	Position prediction	No	No	Yes	No	Yes	Yes	Yes
	Velocity prediction	No	No	Yes	No	Yes	Yes	Yes
	course prediction	No	No	Yes	No	Yes	Yes	Yes
	Dynamic properties	1DOF	No	3DOF	No	2DOF	3DOF	3DOF
Local optimality	COLREGs compliance	High	Med	Med	No	No	High	High
	Path optimality	Yes	No	No	No	Yes	Med	High
	Velocity optimality	Yes	Yes	Yes	Yes	Yes	Med	High
	Course optimality	No	No	Yes	Yes	No	Med	High
	Chattering behaviour overcome	No	No	No	Yes	No	Yes	Yes
	Being stuck overcome	Yes	No	No	Yes	Yes	Yes	Yes
	Blocking goal overcome	No	Yes	Yes	No	Yes	Yes	Yes

(*Continued.*)

Table 1. *tcontinued.*

Method		APF	VO		CAA	DWA	MPC	
Authors		Lyu and Yin	Stenersen	Huang	M.S. Wiig	Eriksen	Kufoalor and Hagen	Tao and Du
Years		2018	2015	2020	2019	2018	2019	This paper
Computational performance	Simulation time step	15 s	0.1 s	1 s	?	1 s	5 s	1 s
	Range of domain	6 nm	300 m	4 nm	2 nm	1,200 m	1,000 m	1,600 m
	Dynamic obstacles	Multi	2	4	1~5 (Sparse)	1	2	3
	Time computation complexity	$O(M)$	$O(N\cdot\beta M)$	$O(\tilde{N}\cdot\beta M)$	$O(M)$	$O\left(N\cdot\frac{T_p}{h}\cdot\beta M\right)$	$O\left(N\cdot\frac{T_p}{h}\cdot\beta M\right)$	$O\left((N-\alpha M)\cdot\frac{T_p}{h}\cdot\beta M\right)$

Notes: symbol '?' means that this subject is not specified or presented clearly in corresponding papers, 'med' is the abbreviation of 'medium'. M, N refer to the number of the static/dynamic obstacles and the number of the candidate velocities. T_p and h refer to the prediction horizon and the discretisation time step for the vessel dynamic model. α and β are parameters used to distinguish the different influence of M. The time computation complexity of the VO algorithm modified by Huang is noted as $O(\tilde{N}\cdot\beta M)$ for the method for solving the quadratic problem is not explained in the paper. The time computation complexity of the method solving the quadratic problem noted as $\tilde{N}$.

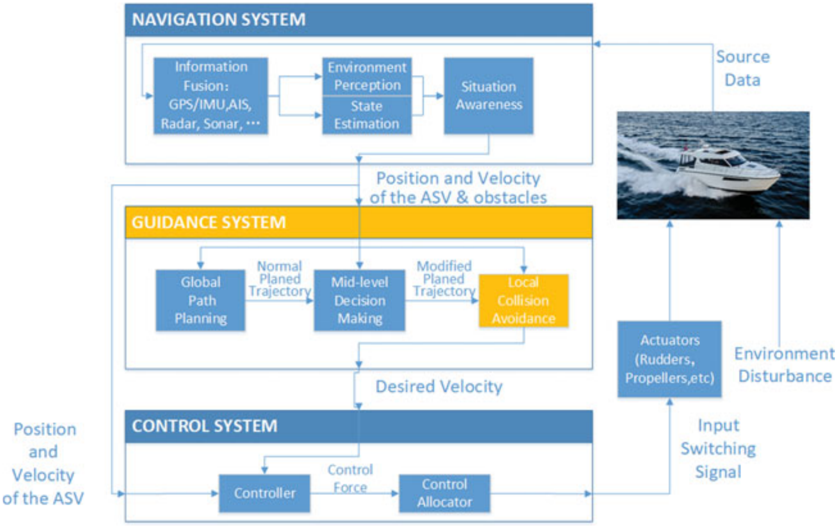


Figure 1. Architecture of the USV GNC system.

Assumption 1: The planned trajectories are known, denoted as

$$\boldsymbol{T_d}(k) = \begin{bmatrix} \boldsymbol{\eta_d}(k) \\ \boldsymbol{\upsilon_d}(k) \end{bmatrix} \tag{1}$$

where k is the index of the waypoints.

Assumption 2: The positions and velocities of obstacles are known when they enter the detection range r_D. The positions of static obstacles are denoted as $(x_i,\ y_i)$ and the positions and velocities of target ships are denoted as $\begin{bmatrix} \boldsymbol{\eta}_j & \boldsymbol{\upsilon}_j \end{bmatrix}^{\mathsf{T}}$, where i and j are the indexes of static obstacles and target ships, respectively.

Assumption 3: The trajectories of target ships are known, denoted as

$$\boldsymbol{T_j}(t) = \begin{bmatrix} \boldsymbol{\eta}_j(t) \\ \boldsymbol{\upsilon}_j(t) \end{bmatrix} \tag{2}$$

where $\boldsymbol{\eta} = \begin{bmatrix} x & y & \psi \end{bmatrix}^{\mathsf{T}}$, x and y are the position coordinates, and ψ is the orientation in north-east-down-fixed inertial frame $\{n\}$. Here, $\boldsymbol{\upsilon}$ is the velocity vector in body-fixed frame $\{b\}$, $\boldsymbol{\upsilon} = \begin{bmatrix} u & v & r \end{bmatrix}^{\mathsf{T}}$, where u, v and r are surge velocity, sway velocity and yaw rate.

2.2. USVs motion model

To accurately express the underactuations and nonlinearities of USVs' motion, a 3DOF mathematical motion model of the USV (Fossen, 2011) is used:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R} \cdot \boldsymbol{\upsilon} \tag{3}$$

$$\boldsymbol{M}\dot{\boldsymbol{\upsilon}} = \boldsymbol{\tau}\boldsymbol{\upsilon} - \boldsymbol{C}(\boldsymbol{\upsilon})\boldsymbol{\upsilon} - \boldsymbol{D}(\boldsymbol{\upsilon})\boldsymbol{\upsilon} \tag{4}$$

Table 2. *Viknes 830 parameters.*

Name	Value	Name	Value
Length (m)	8.5	Max Thrust Force (N)	13,100.0
Width (m)	3	Min Thrust Force (N)	−6,550.0
Mass (kg)	3,980.0	Max Rudder Force (N)	±645.0
Inertia (kg/m^2)	19,703.0	Rudder Length (m)	4.0

Table 3. *Variables and coefficients of the controller.*

Notation	Explanation	Notation	Explanation
τ_z	control outputs	m	mass of the USV
$\tau_{z,\mathrm{c}}$	control laws	I_z	rotational inertia of the USV
$\tau_{z,\min}$	minimum control outputs	l_r	rudder lever arm
$\tau_{z,\max}$	maximum control outputs	$K_{p\psi}$	coefficient of the controller
X_u	coefficient in $\boldsymbol{D}(\boldsymbol{v})$	$T_{d\psi}$	coefficient of the controller
X_{uu}	coefficient in $\boldsymbol{D}(\boldsymbol{v})$	u_d	desired speed
X_{uuu}	coefficient in $\boldsymbol{D}(\boldsymbol{v})$	r_d	desired yaw rate

where $\boldsymbol{M}$, $\boldsymbol{C}(\boldsymbol{\upsilon})$ and $\boldsymbol{D}(\boldsymbol{\upsilon})$ are inertia, Coriolis and damping matrixes. The transformation matrix from $\{b\}$ to $\{n\}$ is given as

$$\boldsymbol{R}(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

The vessel of interest in this study is the Viknes830 USV. The parameters of Viknes 830 are shown in Table 2.

2.3. *Velocity controller*

The COLAV algorithm outputs the desired velocities to the control system which produces switching signals to actuators by controllers and control allocators based on the position and velocity signals. By following the desired velocities, USVs can avoid obstacles and track the planned trajectories.

In this paper, a simple feedback proportional controller is used as an example, and better velocity controllers could be used to improve the performance of the COLAV algorithm in real practice. Considering the input saturation, the control outputs τ_z, $z = 1,\ 2,\ 3$ are defined as

$$\tau_z = \begin{cases} \tau_{z,c} & \text{if } \tau_{z,\min} < \tau_{z,c} < \tau_{z,\max} \\ \tau_{z,\min} & \text{if } \tau_{z,c} \le \tau_{z,\min} \\ \tau_{z,\max} & \text{if } \tau_{z,c} \ge \tau_{z,\max} \end{cases} \tag{6}$$

$$\tau_{z,\mathrm{c}} = \begin{cases} -X_u u - X_{uu}u - X_{uuu}u - mrv + K_{px}(u_d - u), & \textit{when} \quad z = 1 \\ \dfrac{K_{p\psi} I_z T_{d\psi}(r_d - r)}{l_r}, & \textit{when} \quad z = 2 \\ K_{p\psi} I_z T_{d\psi}(r_d - r), & \textit{when} \quad z = 3 \end{cases} \tag{7}$$

The variables and coefficients above are collected in Table 3.

https://doi.org/10.1017/S0373463322000315 Published online by Cambridge University Press

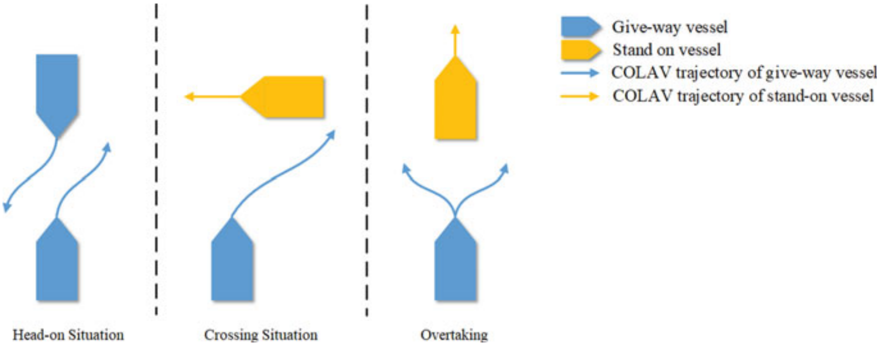


Figure 2. *Encounter situations and COLAV manoeuvre.*

2.4. COLREGs

To avoid traditional vessels at sea, USVs must comply with the COLREGs.

A collision situation is first identified by computing the closest point of approach (CPA), given the current pose of the USV. The vessel is deemed to be in a collision situation if the distance to the closest point of approach (DCPA) d_{CPA} and the time to the closest point of approach (TCPA) t_{CPA} are less than the safe threshold d_{SAFE} and t_{SAFE},

$$0 \leq t_{\mathrm{CPA}} \leq t_{\mathrm{SAFE}} \quad \wedge \quad d_{\mathrm{CPA}} \leq d_{\mathrm{SAFE}} \tag{8}$$

$$t_{\mathrm{CPA}} = \frac{(\boldsymbol{p}_B - \boldsymbol{p}_{\mathrm{A}}) \cdot (\mathbf{v}_{\mathrm{A}} - \mathbf{v}_B)}{||\mathbf{v}_{\mathrm{A}} - \mathbf{v}_B||} \tag{9}$$

$$d_{\mathrm{CPA}} = ||(\boldsymbol{p}_{\mathrm{A}} + \mathbf{v}_{\mathrm{A}} t_{\mathrm{CPA}}) - (\boldsymbol{p}_B + \mathbf{v}_B t_{\mathrm{CPA}})|| \tag{10}$$

where $\boldsymbol{p}_{\mathrm{A}} = (x_{\mathrm{A}}, y_A)$ is the position of the USV's geometry centre, $\boldsymbol{p}_B = (x_B, y_B)$ is the position of the obstacle's geometry centre, and $\mathbf{v}_{\mathrm{A}}$ and $\mathbf{v}_{\mathrm{B}}$ are the velocities of the USV and the obstacle. Thus, when a possible collision situation is present, it remains to identify the applicable COLREGs rules.

Rules 13–17 in COLREGs stipulate the responsibilities of the give-way vessel and the stand-on vessel in the three encounter situations, namely head-on situation, crossing situation, and overtaking, as shown in Figure 2.

These three encounter situations contain five different scenarios determined by the relative bearing of the target vessel to the USV.

$$\beta_{BA} = \arctan 2(y_A - y_B, x_A - x_B\ ) - \ \psi_B \tag{11}$$

The USV acts as the give-way vessel in three scenarios:

(1) head-on, $\beta_{BA} \in [-15°, 15°)$;
(2) the target vessel crossing from starboard side, $\beta_{BA} \in [15°, 112.5°)$;
(3) the target vessel being overtaken by the USV, $\beta_{AB} \in [112.5°, 180°) \cup [-180°, -112.5°)$.

Moreover, the USV acts as the stand-on vessel in two scenarios:

(1) the target vessel crossing from port side, $\beta_{BA} \in [-112.5°, 15°)$;
(2) the target vessel overtaking the USV, $\beta_{BA} \in [112.5°, 180°) \cup [-180°, -112.5°)$.

As target ships sometimes may violate their responsibilities of the give-way vessel for different practical reasons, the USV should also be able to avoid collisions alone. Therefore, communications between the USV and target ships are not necessary in this paper.

https://doi.org/10.1017/S0373463322000315 Published online by Cambridge University Press

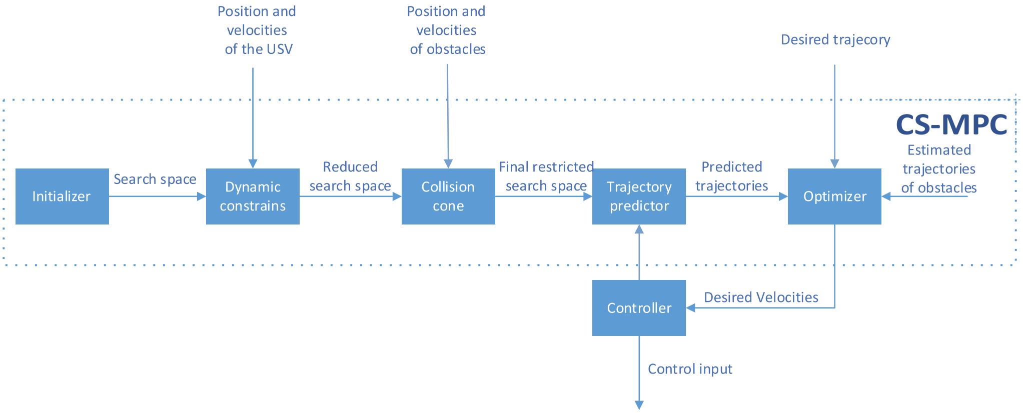


Figure 3. Architecture of the CS-MPC COLAV algorithm.

Remark 1: In two-vessel encounter situations, Rule 17 of COLREGs does not require that the stand-on vessel 'must' or 'have to' always keep its course and speed. The stand-on vessel can also take necessary and understandable manoeuvres to change its course or speed. Since USVs cannot communicate with the target vessel, it is necessary for them to take active, effective and understandable collision avoidance manoeuvres alone.

3. CS-MPC algorithm

The CS-MPC algorithm can produce optimal avoidance manoeuvres in a short runtime to avoid collisions with static and dynamic obstacles while respecting the dynamic constraints of the vessel. The algorithm is based on the MPC and evaluates vessel-feasible trajectories.

Therefore, the CS-MPC algorithm is designed with the short-term perspective of COLREGs in mind, namely situations where the stand-on requirement may need to be ignored to avoid collision in compliance with rule 17.

The CS-MPC algorithm architecture is shown in Figure 3. The initialiser finds the search space of the USV before a voyage. During the voyage, the search space is further trimmed by the dynamic constraints and the collision cone according to the position and velocities of the USV and obstacles. The trajectory predictor produces the trajectories of the velocities in the final restricted search space. The optimiser compares the predicted trajectories, the estimated trajectories of obstacles and the desired trajectory, which can originate from either the upper algorithm of the guidance system or directly from a user, and outputs the desired velocities to the controller based on an objective function.

The CS-MPC algorithm improves the agility of COLAV by modifying the MPC in the following two aspects:

(1) the solutions are closer to the optimal ones for using the dynamic ship motion models. First, the accuracy of the search space is improved by an initialiser considering the nonlinearity of USVs. Second, the accuracy of the trajectory prediction is improved by solving the equations numerically;
(2) the computation complexity is reduced by introducing the collision cone. The dangerous velocities are shielded from the search space using the VO method.

The CS-MPC algorithm is explained in detail as follows.

3.1. Search space

The CS-MPC algorithm makes optimal COLAV decisions based on the hydrodynamic characteristics of USVs, which can be represented as the search space $\boldsymbol{V}_s$. It is a set of discrete velocities with which

https://doi.org/10.1017/S0373463322000315 Published online by Cambridge University Press

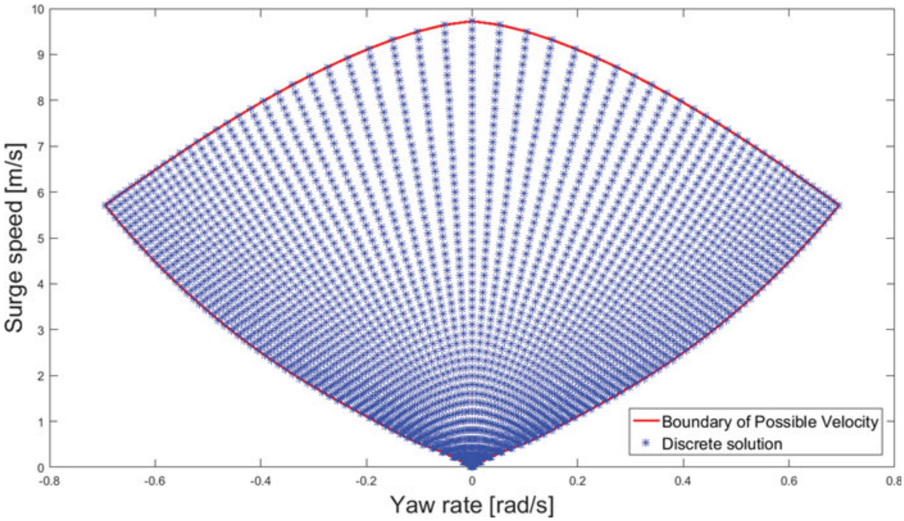


Figure 4. *Search space $\boldsymbol{V}_s$ of the USV.*

the USV can arrive at within the prediction time. To obtain an accurate search space, the USV motion model in Equations (3)–(5) is used, which retains the underactuated characteristics and the nonlinear relationship between velocities and rudder forces. By setting the initial value $\boldsymbol{\nu}_0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$, time step $t_s = 0.1$ s and discretising $\tau_i \in [\tau_{i,\min}, \tau_{i,\max}]$ averagely, after a sufficient number of iterations (in this paper, 100 steps), the numerical solutions of Equations (3)–(5) are obtained with the fourth-order Runge–Kutta method. These solutions constitute the search space of the USV $\boldsymbol{V}_s = \begin{bmatrix} \boldsymbol{u} & \boldsymbol{r} \end{bmatrix}^{\mathsf{T}}$. Putting it more intuitively, the search space can be accurately represented as an irregular pattern, as shown in Figure 4, rather than a simple rectangular space.

Unfortunately, due to the limits of the prediction time T_p, the current velocity $\boldsymbol{\upsilon_a} = (u_a, v_a, r_a)$ and the input saturations of the propeller and rudder, the USV can only arrive at a finite set of velocities $\boldsymbol{V}_d$, which can be defined as follows:

$$\dot{\boldsymbol{\upsilon}}_k = \boldsymbol{M}^{-1}(\boldsymbol{\tau}_k - \boldsymbol{C}(\boldsymbol{\upsilon_a})\boldsymbol{\upsilon_a} - \boldsymbol{D}(\boldsymbol{\upsilon_a})\boldsymbol{\upsilon_a}) \tag{12}$$

$$\boldsymbol{V}_d = \left\{ \begin{array}{c} (u, r) \in R \times R | u \in [u_a + \dot{u}_{\min} T_p,\ u_a + \dot{u}_{\max} T_p\ ] \\ \wedge r \in [r_a + \dot{r}_{\min} T_p,\ r_a + \dot{r}_{\max} T_p] \end{array} \right\} \tag{13}$$

where $k \in \{\min,\ \max\}$. Therefore, the search space is reduced to a subset of $\boldsymbol{V}_s$, called feasible velocities $\boldsymbol{V}_f$,

$$\boldsymbol{V}_f = \{\boldsymbol{V}_d \cap \boldsymbol{V}_s\} \tag{14}$$

For example, in the scenario that the current states of USV are $\boldsymbol{\eta_a} = \left(0\ 0\ 0^{\circ}\right)$, $\boldsymbol{\upsilon_a} = \left(7 \text{ m/s}\ 0\ 0\right)$, and the states of target ship are $\boldsymbol{\eta_b} = \left(800 \text{ m}\ 800 \text{ m}\ -90^{\circ}\right)$, $\boldsymbol{\upsilon_b} = \left(7 \text{ m/s}\ 0\ 0\right)$, the prediction time is $T_p = 2.5$ s, and the resolution of surge speed and yaw rate are 0.2 m/s and 0.01 rad/s. The reduced search space is the intersection area of the blue and red dotted lines shown in Figure 6.

3.2. Collision cone

Another important advantage of the CS-MPC algorithm is the short runtime, which is directly influenced by the scale of the search space. To reduce the runtime, the search space is further reduced by introducing collision cones for static or dynamic obstacles. Given a USV A and an obstacle B, the collision cone takes the position of the USV's geometric centre as its vertex $\boldsymbol{p}_{\mathrm{A}}$, and the two tangent lines of a circle from the

https://doi.org/10.1017/S0373463322000315 Published online by Cambridge University Press

vertex as its edges, denoted as $\boldsymbol{p}_{AB\,\text{left}}^{\perp}$ and $\boldsymbol{p}_{AB\,\text{right}}^{\perp}$. The circle's centre is the position of the obstacle's geometric centre $\boldsymbol{p}_{\text{B}}$, and its radius is the sum of the safe range of the USV r_{A} and the obstacle r_{B}.

Assumption 4: The state of the USV changes immediately. Therefore, the predicted heading of the USV can be calculated by Equations (15), (20) and (21).

$$\tilde{\psi} = \psi + r \cdot T_p \tag{15}$$

Based on Assumption 4, the collision cone can be expressed as follows:

$$\alpha = \arcsin\left(\frac{r_{\text{A}} + r_{\text{B}}}{||\mathbf{p}_{\text{AB}}||}\right) \tag{16}$$

$$\boldsymbol{p}_{\text{AB}\,\text{left}}^{\perp} = \boldsymbol{J}(-\alpha + \pi/2)\frac{\boldsymbol{p}_{\text{AB}}}{||\boldsymbol{p}_{\text{AB}}||} \tag{17}$$

$$\boldsymbol{p}_{\text{AB}\,\text{left}}^{\perp} = \boldsymbol{J}(\alpha - \pi/2)\frac{\boldsymbol{p}_{\text{AB}}}{||\boldsymbol{p}_{\text{AB}}||} \tag{18}$$

$$\boldsymbol{VO}_{\boldsymbol{A}|\boldsymbol{B}} = \{\mathbf{v}_{\text{A}} | \mathbf{v}_{\boldsymbol{BA}} \cdot \boldsymbol{p}_{\text{AB}\,\text{left}}^{\perp} \geq 0 \wedge \mathbf{v}_{\boldsymbol{BA}} \cdot \boldsymbol{p}_{\text{AB}\,\text{right}}^{\perp} \geq 0\} \tag{19}$$

where $\boldsymbol{p}_{\text{AB}}$ is the relative position of the obstacle with respect to the USV $\boldsymbol{p}_{\text{AB}} = \boldsymbol{p}_{\text{B}} - \boldsymbol{p}_{\text{A}}$, likewise, their relative velocity is $\mathbf{v}_{\boldsymbol{BA}} = \mathbf{v}_{\text{A}} - \mathbf{v}_{\text{B}}$, where $\mathbf{v}$ is the velocity vector in the north-east-down-fixed inertial frame $\{n\}$, and u and v are surge velocity and sway velocity as described in Section 2.1. Here, $\boldsymbol{J}(\tilde{\psi})$ is the rotation matrix rotating a matrix with the angle of $\tilde{\psi}$,

$$\mathbf{v} = \boldsymbol{J}(\tilde{\psi})\begin{bmatrix} u & v \end{bmatrix}^{\mathsf{T}} \tag{20}$$

$$\boldsymbol{J}(\tilde{\psi}) = \begin{bmatrix} \cos\tilde{\psi} & -\sin\tilde{\psi} \\ \sin\tilde{\psi} & \cos\tilde{\psi} \end{bmatrix} \tag{21}$$

As the velocities $\mathbf{v}_{\text{A}}$ in $\boldsymbol{VO}_{\boldsymbol{A}|\boldsymbol{B}}$ pointing to the collision cone lead the USV to obstacles or target vessels, it should never be chosen. Consequently, these velocities could be shielded from the search space. The runtime is shortened, because the velocities in the shielded search space will not be evaluated in the following steps. For the example in Section 3.1, the collision cone can be visually represented by Figure 5, and the velocities in the final restricted search space are shown in Figure 6 as blue stars. To prevent too many feasible velocities from being shielded, the collision cone should only be calculated when the distance between the USV and an obstacle is less than the safe range, called the collision cone range r_C in this paper.

3.3. *Trajectory prediction*

From the final restricted search space, the optimal solution can be selected based on the situation in a prediction time. Therefore, the trajectory prediction should be as accurate as possible. Meanwhile, to reduce the computation complexity, the small change assumption is made as follows.

Assumption 5: If the length of the iteration step is short enough, the Coriolis matrixes $\boldsymbol{C}(\boldsymbol{v})$ and damping matrixes $\boldsymbol{D}(\boldsymbol{v})$ are constants within the iteration step.

Then the predicted trajectories within a prediction time T_p are obtained by solving the 3DOF mathematic motion model of USVs using Equations (3)–(5) numerically with the improved Euler

https://doi.org/10.1017/S0373463322000315 Published online by Cambridge University Press

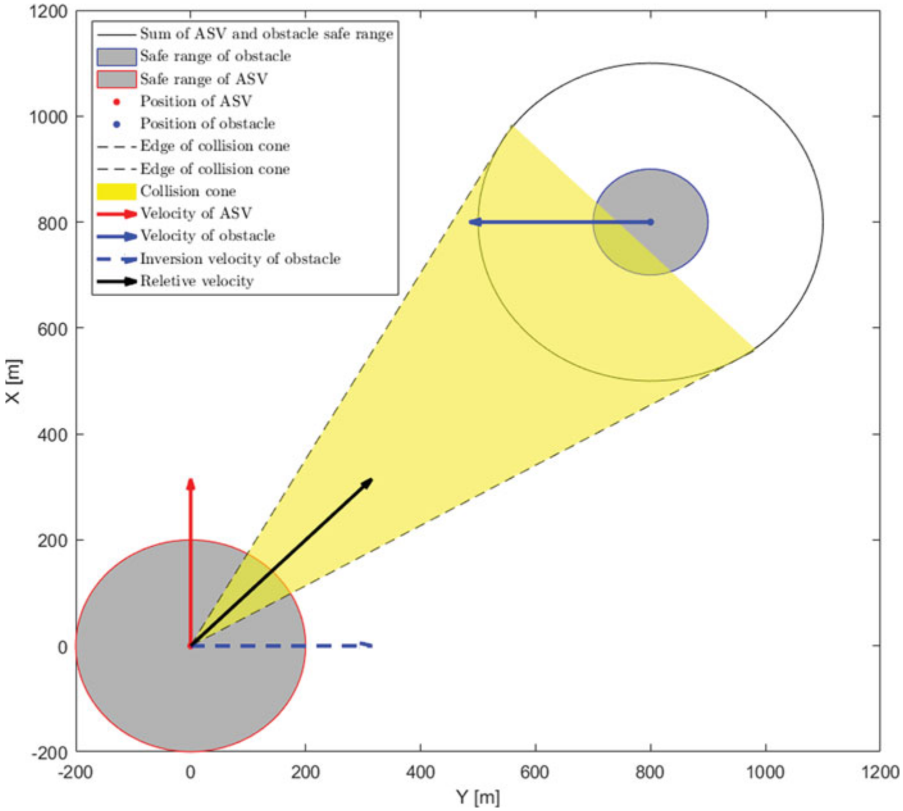


Figure 5. *Schematic diagram of collision cone.*

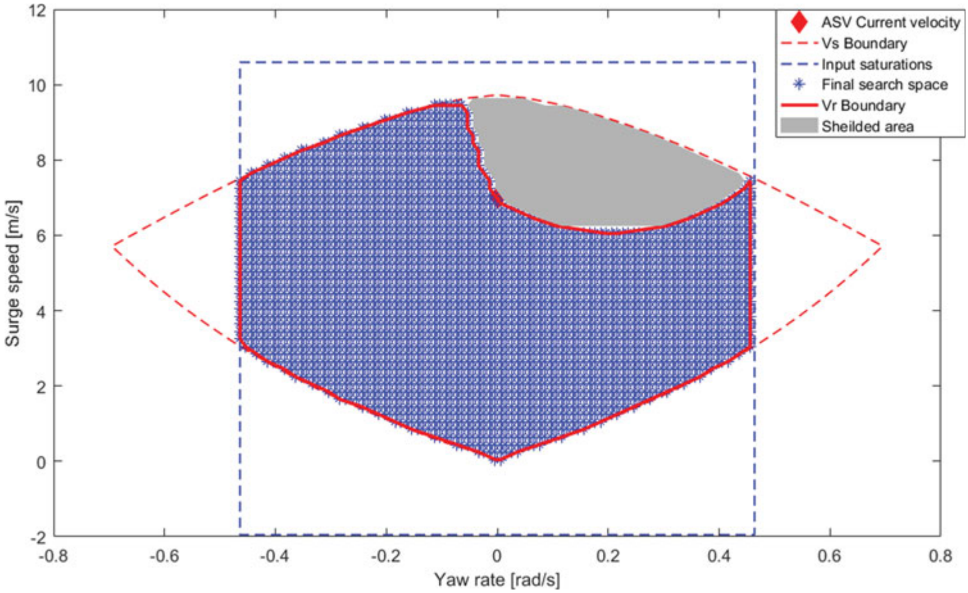


Figure 6. *Search space of CS-MPC.*

https://doi.org/10.1017/S0373463322000315 Published online by Cambridge University Press

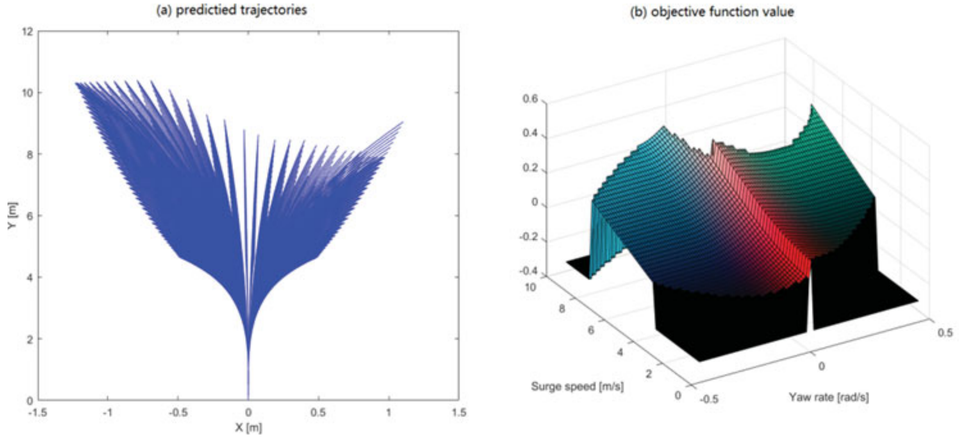


Figure 7. *(a) predicted trajectories of $\boldsymbol{V}_r$. (b) Objective function value of $\boldsymbol{V}_r$.*

method,

$$\upsilon_p = \boldsymbol{\upsilon}_n + h\boldsymbol{M}^{-1}(\boldsymbol{\tau} - \boldsymbol{C}(\boldsymbol{\upsilon}_n)\boldsymbol{\upsilon}_n - \boldsymbol{D}(\boldsymbol{\upsilon}_n)\boldsymbol{\upsilon}_n) \tag{22}$$

$$\upsilon_c = \boldsymbol{\upsilon}_n + h\boldsymbol{M}^{-1}(\boldsymbol{\tau} - \boldsymbol{C}(\boldsymbol{\upsilon}_n)\boldsymbol{\upsilon}_p - \boldsymbol{D}(\boldsymbol{\upsilon}_n)\boldsymbol{\upsilon}_p) \tag{23}$$

$$\upsilon_{n+1} = \frac{1}{2}(\upsilon_c + \upsilon_p) \tag{24}$$

$$\boldsymbol{\eta}_{n+1} = \boldsymbol{\eta}_n + \frac{h}{2}\boldsymbol{R}(\upsilon_n + \upsilon_{n+1}) \tag{25}$$

where n is the index of iteration steps, h is the length of the prediction time step, and $\boldsymbol{\tau} = (\tau_z | z = 1,\ 2,\ 3)$ is the current control force calculated from Equations (6) and (7).

The predicted trajectories for the example in Section 3.1 are shown in Figure 7(a).

3.4. Selecting the desired velocities

Based on the predicted trajectories, the candidate velocities in the final restricted search space can be evaluated by a criterion. Assuming that in the time step k there are N candidate velocities in $\boldsymbol{V}_r$ and M obstacles around the USV, the criterion can be represented as the objective function, shown as

$$(u_r, r_r) = \arg\max \mathcal{H}^k \tag{26}$$

$$\mathcal{H}^k = \begin{cases} q_\chi \cdot \dfrac{|\chi - \chi_{\text{goal}}|}{\max\limits_{i\in N} |\chi - \chi_{\text{goal}}|} + q_d \cdot \dfrac{d_{OA}}{\max\limits_{i\in N} \ d_{OA}} + q_u \cdot \dfrac{|u|}{\max\limits_{i\in N} |u|} + q_{\mathcal{T}} \cdot T, & d^k_{i,n} > d_{\text{SAFE}} \\ -1, & d^k_{i,n} \le d_{\text{SAFE}} \end{cases} \tag{27}$$

$$d^k_{i,n} = \min_{m\in M} ||\boldsymbol{p}^{\boldsymbol{k}}_{\boldsymbol{A},\boldsymbol{n}} - \boldsymbol{p}^{\boldsymbol{k}}_{\boldsymbol{B},\boldsymbol{m}}|| \tag{28}$$

where (u_r, r_r) is regarded as the optimal solution, $q_\chi < 0$, $q_u > 0$, $q_d > 0$, $q_{\mathcal{T}} < 0$ are coefficients tuned in different missions, and $d^k_{i,n}$ is the minimum distance between the nth discrete position in the ith predicted trajectory of the USV $\boldsymbol{p}^{\boldsymbol{k}}_{\boldsymbol{A},\boldsymbol{n}}$ and the predicted positions of the mth obstacles $\boldsymbol{p}^{\boldsymbol{k}}_{\boldsymbol{B},\boldsymbol{m}}$. All other variables take the values at the end of each trajectory. The first term ensures the course directing to the destination, where χ, χ_{goal} are the predicted course and the course toward the goal. The second term keeps the USV far away from obstacles, where d_{OA} represents the sum of the distances between the USV and obstacles. The third term is to make the USV maintain a high surge speed u. The last term presents the COLREGs penalty, evaluates whether the avoidance manoeuvre complies with the

https://doi.org/10.1017/S0373463322000315 Published online by Cambridge University Press

COLREGs and maintains a consistent avoidance manoeuvre. Here, $\mathcal{T}$ is the Boolean variable, whose value is true (value ‘1’) if the following conditions are all met.

(1) The USV is in a collision situation with another vessel. The collision situation is determined in Section 2.4.
(2) The COLAV decision violates COLREGs.

Examining whether the USV will pass the other vessel on its port or starboard side is an efficient technique to determine whether the COLAV decision violates COLREGs. As shown in Figure 2, the USV shall turn to starboard passing the other vessel on its port side in scenarios where the target ship is coming from head or crossing from the starboard side. Moreover, the USV shall not turn to port when it has to take action as will best aid to avoid collision. According to the above, turning to port violates COLREGs in these scenarios. It can be described as

$$\boldsymbol{p}_{\mathbf{AB}} \times \mathbf{v}_{\boldsymbol{BA}} < 0 \tag{29}$$

(3) The COLAV decision is different from the previous avoidance manoeuvre.

As shown in Figure 2, when the USV is overtaking the other vessel, the USV should keep out of the way of the other vessel and could pass it from either side. Similarly, when the USV is being overtaken by another vessel, it could let the other vessel pass from either side. In the above two scenarios, the USV will chatter when it is close to the other vessel, because it could choose to pass the vessel from different sides at time step k and k + 1. To solve this problem, the COLAV decision shall be consistent by adding the penalty when

$$(\boldsymbol{p}_{\boldsymbol{AB},\boldsymbol{k}} \times \boldsymbol{v}_{\boldsymbol{BA},\boldsymbol{k}}) \cdot (\boldsymbol{p}_{\boldsymbol{AB},\boldsymbol{k}+1} \times \boldsymbol{v}_{\boldsymbol{BA},\boldsymbol{k}+1}) < 0 \tag{30}$$

(4) The collision situation has not cleared yet.

To prevent the second-time risk of collision, the COLREGs rule 8 requires that the effectiveness of the action shall be carefully checked until the other vessel is finally past and clear. As a result, the penalty still should be added when the COLAV decision violates COLREGs, even though the action to avoid collision makes $t_{\mathrm{CPA}} \geq \ t_{\mathrm{SAFE;}}$ or $d_{\mathrm{CPA}} \geq d_{\mathrm{SAFE}}$, until the bearing β of the other vessel crosses 22.5 degrees abaft the beam of the USV. At time steps k and $k+1$, this condition can be described as

$$\begin{aligned} &[\beta - (\psi_k + 112^\circ)] \cdot [\beta - (\psi_{k+1} + 112^\circ)] > 0 \\ \wedge\ &[\beta - (\psi_k - 112^\circ)] \cdot [\beta - (\psi_{k+1} - 112^\circ)] > 0 \end{aligned} \tag{31}$$

The objective function values for the above example are shown in Figure 7(b). The RGB colour represents the values of the first three terms in Equation (6). Red represents the values of the course term. Green represents the values of the distance term. Blue represents the values of the surge speed term. In addition, as turning port side violates the COLREGs, all the objective function values of these solutions are lower.

Remark 2: The VO method needs to meet the assumption that USVs move at any constant velocity throughout the avoidance manoeuvre. Therefore, the resulting trajectory is linear (straight line movement). However, this assumption cannot be met when predicting the trajectory using the dynamic motion model of USVs. As a result, there are deviations between the two types of trajectories. The comparison between these two types of trajectories of velocities in the collision cone of Figure 6 is illustrated in Figure 8(a). For the above reasons, two issues arise. First, a small number of shielded velocities may move outside the collision cone, generating the optimal collision avoidance trajectory. Second, a small number of velocities from the final restricted search area may fall into the collision cone along nonlinear trajectories.

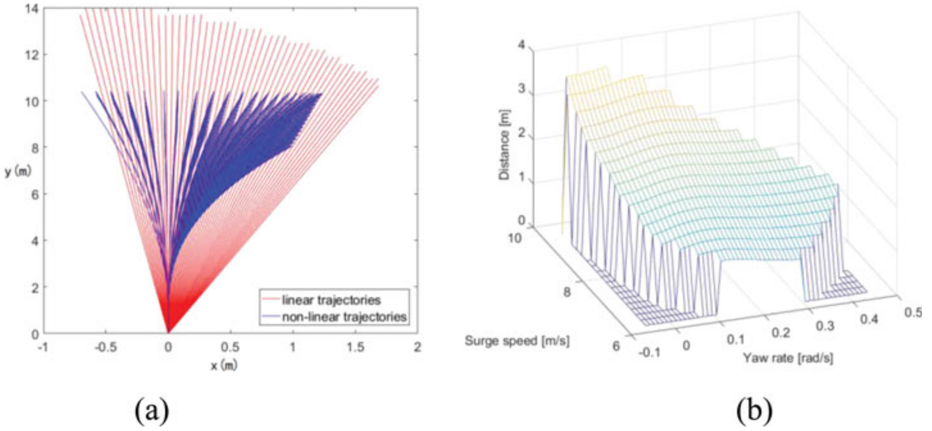


Figure 8. *(a) Comparison between linear and nonlinear trajectories. (b) Deviation between the two types of trajectories.*

For the issue of shielding the optimal trajectories, it should be accepted that in some cases, the solution may be suboptimal. However, for the robustness of the MPC algorithm, suboptimal solutions can still be used and be efficient to avoid collisions.

(1) The deviation between the optimum and suboptimal solutions is acceptable and can be ignored. The distance between the end positions of linear and nonlinear trajectories are shown in Figure 8(b). Therefore, if the optimal solution is in a collision cone, it will be near the boundary of the collision cone. In addition, as the local COLAV time step is very short, it will barely affect the COLAV effectiveness.

(2) Because the CS-MPC uses a rolling optimisation strategy, the low-level controller will follow the desired velocity only at the next time step. In most velocity controllers, such as the one used in this paper, the control forces based on optimal and suboptimal solutions for COLAV are almost the same in the early period when the error appears.

(3) If the optimal solution is in the collision cone, the suboptimal solution will be located on the same side of the collision cone. Additionally, the position of the USV predicted by the suboptimal solution is further away from the collision cone. It means the suboptimal solution has a larger DCPA than the optimal solution. Therefore, if the optimal solution is shielded in a collision cone, the chosen suboptimal solution will be safe but more conservative.

For the issue of some velocities outside the collision cone still being dangerous, the objective function values of these velocities are set to '−1' as in Equations (26)–(28). If there are other feasible velocities, these dangerous velocities will not be chosen.

The pseudo-code of the CS-MPC COLAV algorithm is shown as follows.

```
Algorithm    CS-MPC COLAV algorithm
Input parameters of the USV
Compute the search space $\boldsymbol{V}_s$
While $||P_{\mathrm{ASV}} - P_{\mathrm{goal}}||_2 < C$ do:
Compute the set of dynamically feasible velocity pairs $\boldsymbol{V}_f$
Compute the collision cone $\boldsymbol{VO}_{A|B}$
Find the final restricted search space $\boldsymbol{V}_r = \boldsymbol{V}_f \cap \boldsymbol{V}_s - \boldsymbol{VO}_{A|B}$
Predict the USV trajectories for the velocity pairs in $\boldsymbol{V}_r$
Compute the penalty of the velocity pairs in $\boldsymbol{V}_r$ if violate the COLREGs
```

https://doi.org/10.1017/S0373463322000315 Published online by Cambridge University Press

***Table 4.** Coefficients of the CS-MPC algorithm.*

Name	Value	Name	Value
Simulating time step t_s	1 s	$K_{p\psi}$	12
Prediction time T_p	10 s	$T_{d\psi}$	2
Prediction time step h	0.1 s	Detection range r_D	400 m
q_χ	−0.2	Safe DCPA d_{SAFE}	100 m
q_d	0.4	Safe TCPA t_{SAFE}	50 s
q_u	0.4	Collision cone range r_C	160 m
$q_{\mathcal{T}}$	−0.1	Safe range of USV r_{A}	50 m
K_{px}	1		

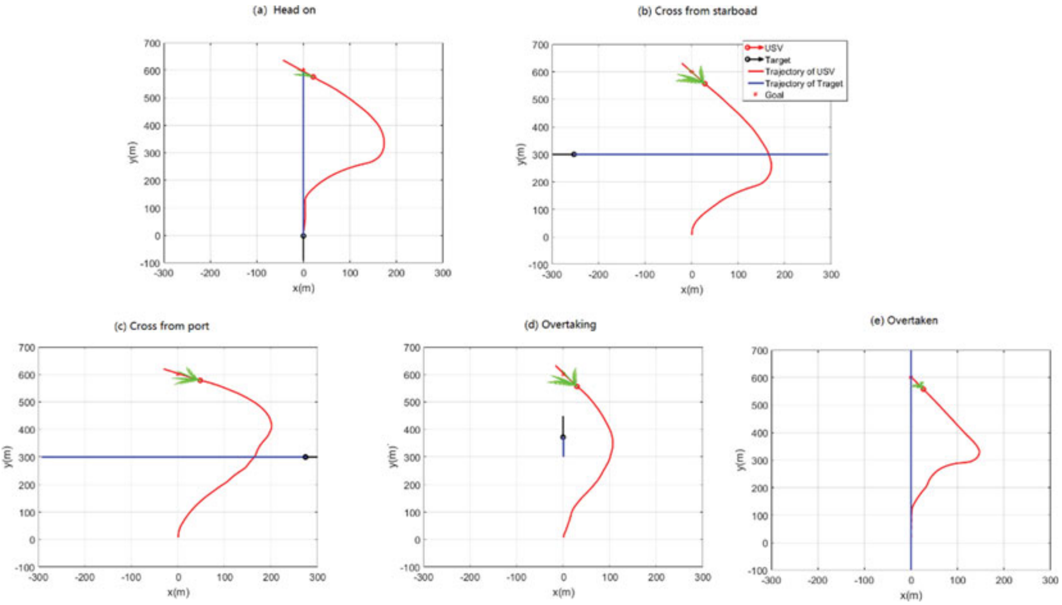


***Figure 9.** Simulation results of COLAV with a single vessel in uncoordinated manoeuvre scenarios.*

```
Evaluate the objective function for the velocity pairs in $\boldsymbol{V_r}$
Numerically search for the maximum value of the objective function, and apply the corresponding velocity pair in $\boldsymbol{V_r}$ as the algorithm output $(u_r, r_r)$
end
```

4. Simulation results

The effectiveness and applicability of the CS-MPC COLAV algorithm were verified by simulation experiments in situations of encountering with a single vessel and multiple vessels. The agility improvement of the CS-MPC was verified by simulation experiments in a more complicated situation, compared with the scenario-based MPC (SB-MPC, Kufoalor et al., 2019). A Viknes 830 USV was selected as the research vessel, whose parameters are shown in Table 2. The coefficients of the CS-MPC algorithm were tuned as in Table 4. These experiments have been simulated with a 2.2 GHz i7-8750 CPU and a 16 GB RAM in a single process.

Table 5. *Simulation conditions of COLAV with single vessel in uncoordinatedly manoeuvre scenarios.*

	USV Initial State		Target Vessel State		Goal	Mean Runtime(ms)	
Situation Group	η_{USV}	υ_{USV}	η_{Tar}	υ_{Tar}	$[N, E]$	SB-MPC	CS-MPC
Head On	[0,0,0°]	[7,0,0]	[0,600,180°]	[7,0,0]	[0,600]	532	423
Cross from Left			[−300,300,90°]			526	452
Cross from Right			[300,300, −90°]			542	445
Overtaking			[0,300,0°]	[1,0,0]		538	417
Overtaken			[0, −100,0°]	[14,0,0]		548	402

Table 6. *Simulation conditions of COLAV with single vessel in coordinated manoeuvre situations.*

	USV 1			USV 2		
Situation Group	η_{USV}	υ_{USV}	Goal	η_{Tar}	υ_{Tar}	Goal
Head On	[0, 0, 0°]	[7, 0, 0]	[0, 600]	[0, 600, 180°]	[7, 0, 0]	[0, 0]
Cross	[0, 0, 0°]	[7, 0, 0]	[0, 600]	[300, 300, 270°]	[7, 0, 0]	[−300, 300]
Overtaking	[0, 100, 0°]	[3, 0, 0]	[0, 500]	[0, −100, 0°]	[10, 0, 0]	[0, 700]

4.1. Encountering a single vessel

Encountering a single vessel is a common situation for ships' COLAV, especially on ocean voyages. As a typical simulation experiment, it can verify the effectiveness of the COLAV algorithm. Assume that the USV is sailing from one waypoint (the initial state) to another waypoint (the destination) in the planned trajectory without disturbance of wind, current and wave.

(1) In the first group of simulations, the USV needs to avoid collision with the target vessel, which does not require coordinated manoeuvres. The simulations are set based on the five scenarios described in Section 2.4. The experimental conditions are shown in Table 5 and the results are shown in Figure 9.

Analysing the experimental results, the runtimes of all iterative steps were within 460 ms, much shorter than the sample time. Therefore, the CS-MPC COLAV algorithm can meet the real-time calculation requirements. In addition, the USV avoided the target vessel, complying with the COLREGs. The USV was able to avoid other vessels even if they did not comply with the COLREGs. Meanwhile, the USV did not oscillate during the COLAV procedures.

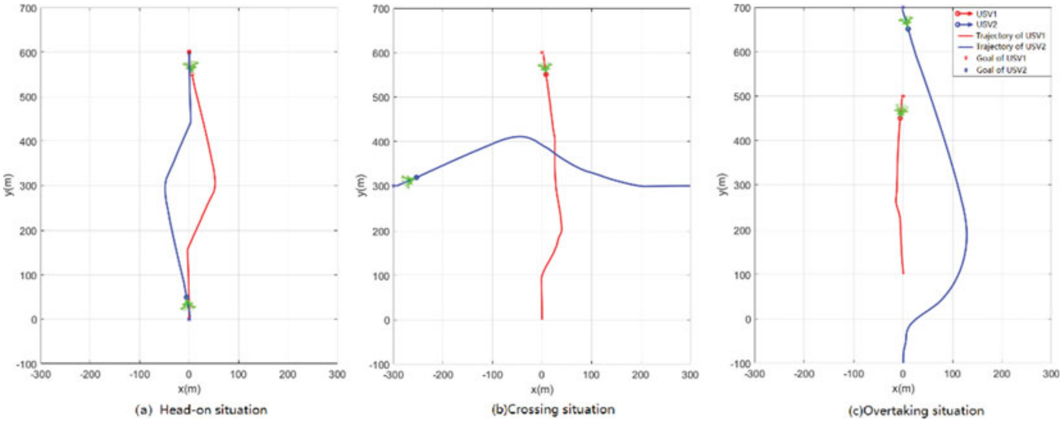
(2) In the second group of simulations, both the USV and the target vessel manoeuvre coordinately with the CS-MPC COLAV algorithm. So, both vessels are regarded as USVs. The simulations are set based on the three encounter situations shown in Figure 2. The experimental conditions are shown in Table 6 and the results are shown in Figure 10.

Analysing the experimental results, even though there is no communication between the two USVs, both vessels manoeuvre coordinately to avoid collision, complying with the COLREGs. In addition, each COLAV manoeuvre is slighter than the one in uncoordinated manoeuvre scenarios.

4.2. Encountering multiple vessels

In coastal navigation, USVs are often in multi-vessel encounter situations, requiring the COLAV algorithm to make intelligent decisions.

https://doi.org/10.1017/S0373463322000315 Published online by Cambridge University Press



(a) Head-on situation

(b)Crossing situation

(c)Overtaking situation

Figure 10. *Simulation results of COLAV with a single vessel in coordinated manoeuvre situations.*

(1) Taking the USV encounter with two target vessels as an example, simulation experiments were carried out. The simulation conditions in Table 5 were combined to form six groups of encounter situations. The results are shown in Figure 11.

From the results, it can be observed that the USV selected effective COLAV decisions in the two-vessel encounter situations. When the COLREGs were impossible to comply with by both vessels [such as the results in Figure 11(c)], the USV even deviated from the COLREGs rules, so as to select the safest manoeuvre.

(2) It is difficult for a USV to avoid collision when it encounters more than two target vessels. IMAZU from Tokyo University of Marine Science and Technology raised many encounter situations that were difficult for multi-vessel COLAV (called the IMAZU problem). To further verify the effectiveness of the CS-MPC algorithm, 11 relatively difficult situations of the IMAZU problem are simulated, where the USV encounters three target vessels at the same time. Considering the difficulty of the IMAZU problem, every vessel should manoeuvre to help avoid collision based on the CS-MPC algorithm. Therefore, all the four vessels in the simulation were regarded as USVs. The simulation conditions are shown in Table 7, and the results are shown in Figure 12.

The simulation results show that the USVs can avoid collision when they encounter with three other target vessels. As there is a collision risk between either of the vessels, the prerequisite for the success of the COLAV is that every vessel should manoeuvre to avoid collision. However, in some scenarios, although the COLAV is achieved, the trajectories of the USVs may be very winding, such as in the experiments shown in Figures 12(g), 12(j) and 12(k).

4.3. Complicated situation

To further verify the superiority in agility of the CS-MPC algorithm, simulations were carried out in a more complicated situation, where the USV should avoid several static obstacles and two target ships. The CS-MPC algorithm was compared with the SB-MPC algorithm. The experiment conditions and some parameters of the SB-MPC algorithm are shown in Table 8. Other parameters of the SB-MPC are the same as the CS-MPC shown in Table 4. The simulation results are shown in Figure 13.

The USV can avoid the static obstacles and target vessels, guided by either the CS-MPC or SB-MPC algorithm.

From Figure 13(c), the CS-MPS mean runtime is 442.2 ms, which is 9.21% less than the SB-MPC mean runtime of 487.1 ms. Especially in period ①, the CS-MPC mean runtime is approximately 250 ms,

https://doi.org/10.1017/S0373463322000315 Published online by Cambridge University Press

Table 7. *Simulation conditions of four vessels encounter situations of IMAZU problem.*

	USV 1			USV 2			USV 3			USV 4		
No.	$\boldsymbol{\eta}$	u (m/s)	Goal	$\boldsymbol{\eta}$	u (m/s)	Goal	$\boldsymbol{\eta}$	u (m/s)	Goal	$\boldsymbol{\eta}$	u (m/s)	Goal
a	[0, 0, 0°]	7	[0, 600]	[0, 700, 180°]	7	[0, 100]	[100, 0, 345°]	7	[−100, 700]	[0, 600, 315°]	7	[−200, 600]
b	[0, 0, 0°]	7	[0, 600]	[0, 700,180°]	7	[0, 100]	[−150, 0, 15°]	7	[100, 700]	[−250, 100, 30°]	7	[200, 600]
c	[0, 0, 0°]	7	[0, 600]	[350, 350,270°]	7	[−350, 350]	[100, 0, 345°]	7	[−100, 700]	[250, 100, 315°]	7	[−250, 600]
d	[0, 0, 0°]	3.5	[0, 600]	[350,350,270°]	7	[−350, 350]	[0, −100, 0°]	7	[0, 700]	[175, 175, 315°]	7	[−250, 600]
e	[0, 0, 0°]	7	[0, 600]	[350, 350,270°]	7	[−350, 350]	[−350, 350,90°]	7	[350, 350]	[−175, 175, 45°]	7	[250, 600]
f	[0, 0, 0°]	3.5	[0, 600]	[0, −100, 0°]	7	[0, 700]	[−100,−100,30°]	7	[100, 600]	[175, 175, 315°]	7	[−250, 600]
g	[0, 0, 0°]	7	[0, 600]	[100, 0, 345°]	7	[−100, 650]	[350, 700,225°]	7	[−375, 0]	[175, 100, 330°]	7	[−250, 600]
h	[0, 0, 0°]	7	[0, 700]	[−150, 0, 15°]	7	[150, 700]	[375, 700,225°]	7	[−375, 0]	[175, 0, 340°]	7	[−250, 600]
i	[0, 0, 0°]	3.5	[0, 600]	[0, −100, 0°]	7	[0, 700]	[375, 375,270°]	7	[−375, 375]	[100, −100, 345°]	7	[−250, 600]
j	[0, 0, 0°]	7	[0, 700]	[−150, 0, 15°]	7	[150, 700]	[375, 350,270°]	7	[−375, 350]	[175, 0, 340°]	7	[−250, 600]
k	[0, 0, 0°]	3.5	[0, 600]	[0, −100, 0°]	7	[0, 700]	[375, 375,270°]	7	[−375, 375]	[100, −50, 345°]	7	[−150, 650]

https://doi.org/10.1017/S0373463322000315 Published online by Cambridge University Press

Table 8. *Simulation conditions of the complicated situation and parameters of the SB-MPC algorithm.*

USV Initial State		Target Vessel State		Static Obstacles	Waypoint	SB-MPC parameters			
$\boldsymbol{\eta}_{\mathbf{USV}}$	$\boldsymbol{\upsilon}_{\mathbf{USV}}$	$\boldsymbol{\eta}_{\mathbf{Tar}}$	$\boldsymbol{\upsilon}_{\mathbf{Tar}}$	$[\boldsymbol{E}, \boldsymbol{N}]$	$[\boldsymbol{E}, \boldsymbol{N}]$	K_I^{COLL}	k_P	k_χ	k_i
[0,0,0°]	[7,0,0]	[−100,300,90°]	[1,0,0]	[0,200]	$\begin{bmatrix} 600, & 600 \\ -600, & 600 \\ -600, & 0 \end{bmatrix}$				
		[300,300, 220°]	[7,0,0]	[−600,300] [−600,400] [400,200] [500,400]		0.4	0.4	–0.2	–0.1

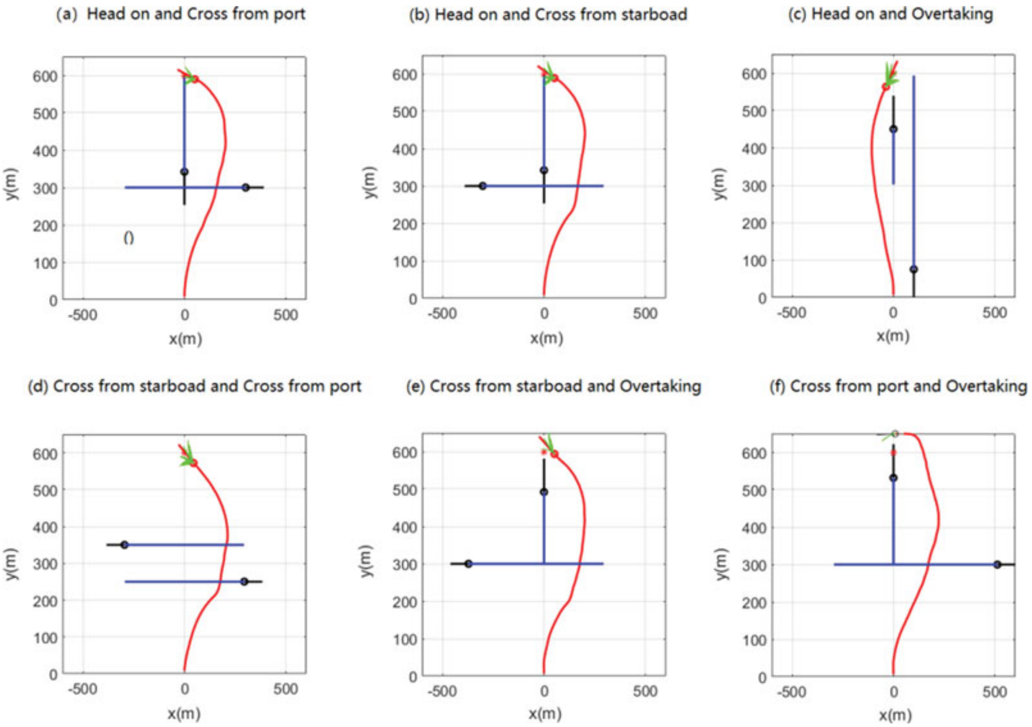


Figure 11. Simulation results of encounter in multi-vessel situations.

because parts of the search space were shielded by both the static obstacle and the target vessel. Therefore, the USV reacted more quickly, guided by the CS-MPC.

From Figure 13(b), the minimum distance from the USV to obstacles (including target vessels) of the CS-MPC was 143.9 m, which was 35.88% larger than that of the SB-MPC at 105.9 m. Therefore, the USV can avoid obstacles by making safer COLAV decisions guided by the CS-MPC.

From Figure 13(a), the USV guided by the CS-MPC took action earlier in the procedures of both COLAV and returning to the planned trajectory. In addition, the COLAV manoeuvre of the USV guided by the CS-MPC is more obvious. Above all, the COLAV decision of the CS-MPC is better according to the COLREGs rules.

5. Discussion

The proposed CS-MPC COLAV algorithm guides the USV more quickly with better COLAV decisions, which means that it improves the USV's COLAV agility. However, the CS-MPC will not lead the USV precisely to the waypoint, if there are obstacles or target vessels near the waypoint.

The CS-MPC runtime of the worst-case scenario is in the scenarios where there are no static obstacles and target vessels within the collision cone range $r_c = 160$ m, because there are no collision cones to shield the search space. From Figure 13(c), the runtimes of the CS-MPC are almost equal to those of the SB-MPC in some periods. It is because in these periods, the search space is not shielded by collision cones and the numbers of the candidate manoeuvres for both algorithms are initially set to equal. As the worst-case runtime of the CS-MPC is lower than the simulation time step, it could be used in a real-time COLAV.

However, guided by the CS-MPC algorithm, the USV sometimes may not precisely reach the waypoint, just like the results in some simulations above. It is because the velocities leading to the waypoint are in the collision cone. If the USV precisely reaches the waypoint, it may collide with the target ship

https://doi.org/10.1017/S0373463322000315 Published online by Cambridge University Press

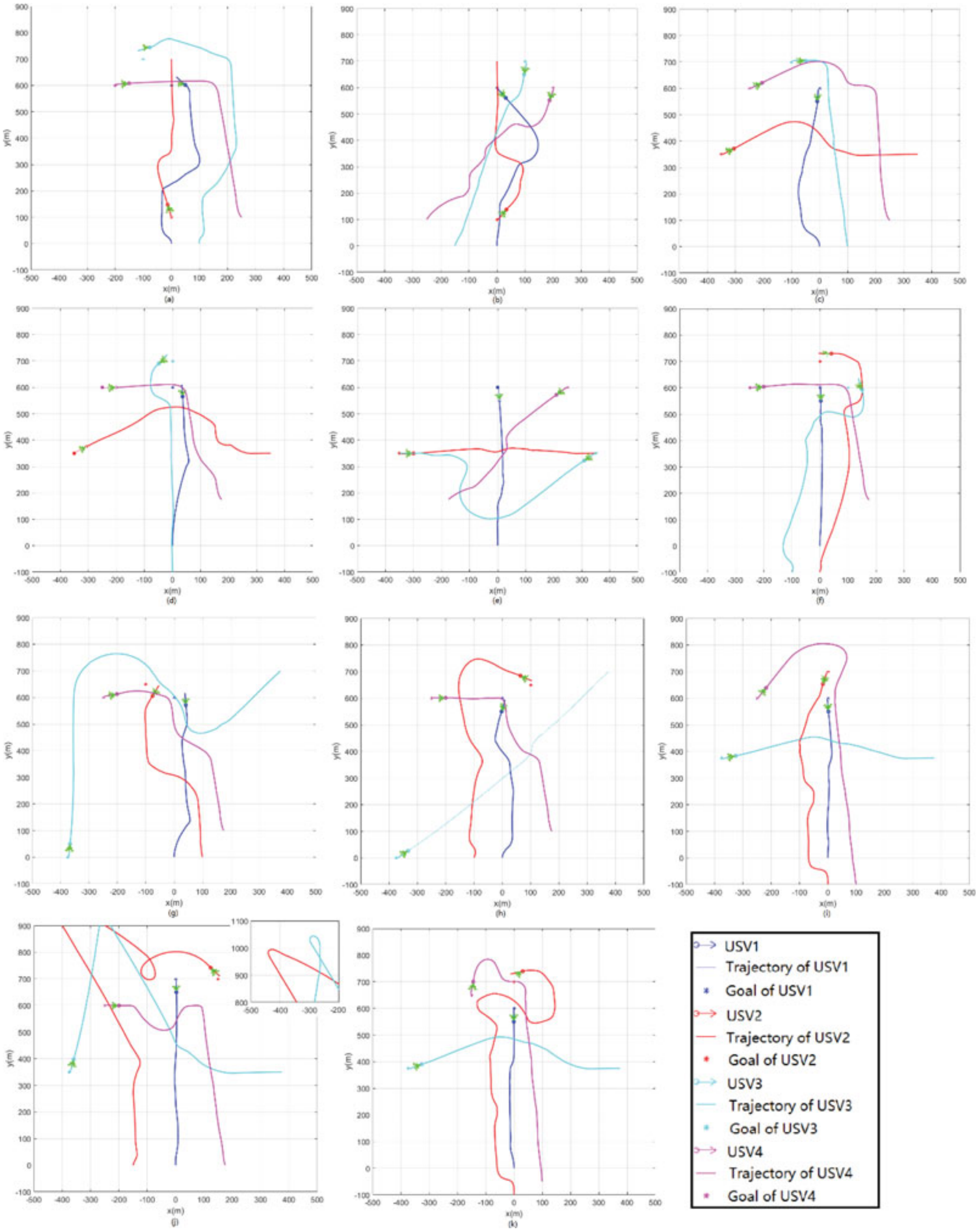


Figure 12. *COLAV simulation results for four vessels in the IMAZU problem.*

afterwards or have to make an emergency stop manoeuvre. Fortunately, this problem will not affect the autonomous navigation for the USV, which is combined with a high-level path planning system. As a result of the complicated situation depicted in Figure 13(a), the next one on the planned path would replace the waypoint, and the USV would navigate towards the new waypoint with a better COLAV manoeuvre.

https://doi.org/10.1017/S0373463322000315 Published online by Cambridge University Press

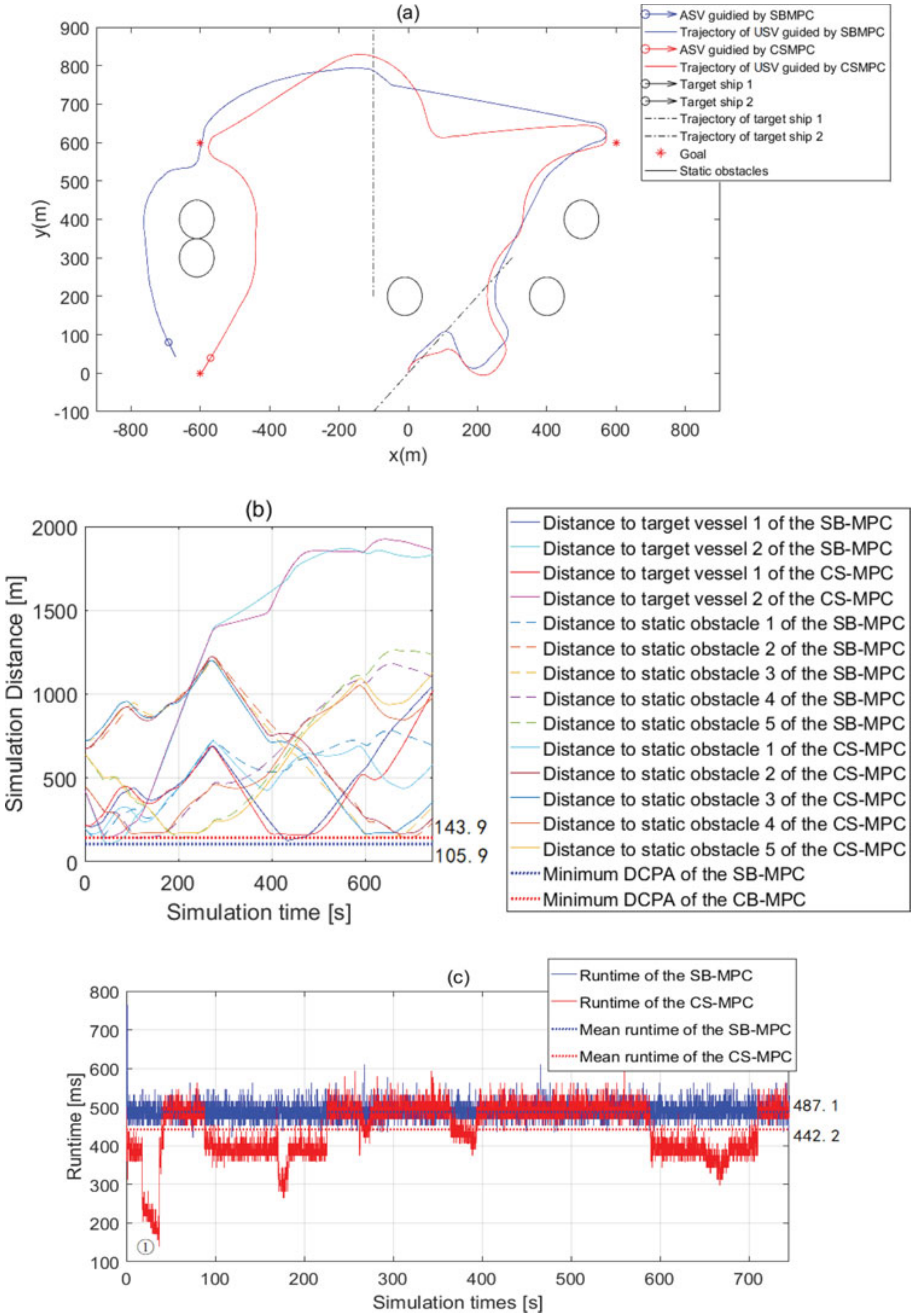


Figure 13. *Experiment results in the complicated situation.*

6. Conclusion

In this paper, a collision shielded MPC COLAV algorithm is proposed for USVs, which improves the agility of the COLAV. The optimality of the COLAV decision is improved by an accurate trajectory prediction considering the nonlinear dynamics of the underactuated USVs and a modified objective

https://doi.org/10.1017/S0373463322000315 Published online by Cambridge University Press

function based on COLREGs. In addition, the algorithm runtime is reduced by shielding the dangerous velocities in the search space with the collision cone, prior to the trajectory prediction. The effectiveness, applicability and the agility improvement are verified by the simulation experiments in single-vessel, multi-vessel encounter situations and a complicated situation.

As the CS-MPC can also guide USVs to destinations, by combining the CS-MPC with velocity controllers, it can be used not only in the COLAV subsystem, but also for trajectory tracking in the guidance system. The USVs' safely autonomous navigations in complicated waters can be achieved.

References

Benjamin, M. R., Leonard, J. J., Curcio, J. A. and Newman, P. M. (2006). A method for protocol-based collision avoidance between autonomous marine surface craft. *Journal of Field Robotics*, **23**(5), 333–346.

Brock, O. and Khatib, O. (1999). High-speed Navigation Using the Global Dynamic Window Approach. *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, Detroit, USA.

Chiang, H. L. and Tapia, L. (2018). COLREG-RRT: An RRT-based COLREGS-compliant motion planner for surface vehicle navigation. *IEEE Robotics and Automation Letters*, **3**, 2024–2031.

Enevoldsen, T. T., Reinartz, C. and Galeazzi, R. (2021). COLREGs-Informed RRT* for Collision Avoidance of Marine Crafts. *Proceedings of 2021 IEEE International Conference on Robotics and Automation*, Xi'an, China.

Eriksen, B. O. H. (2019). *Collision avoidance and motion control for autonomous surface vehicles*. Ph.D. Thesis, Norwegian University of Science and Technology.

Eriksen, B. O. H, Wilthil, E. F., Flaten, A. L., Brekke, E. F. and Breivik, M. (2018). Radar-based Maritime Collision Avoidance Using Dynamic Window. *Proceedings of the 2018 IEEE Aerospace Conference*, Big Sky, USA.

Fiorini, P. and Shiller, Z. (1998). Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research*, **17**(7), 760–772.

Fossen, T. I. (2011). *Handbook of Marine Craft Hydrodynamics and Motion Control*. United Kingdom: John Wiley & Sons, Ltd. Chichester.

Fox, D., Burgard, W. and Thrun, S. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, **4**(1), 23–33.

Hagen, I. B., Kufoalor, D. K. M., Brekke, E. F. and Johansen, T. A. (2018). MPC-based Collision Avoidance Strategy for Existing Marine Vessel Guidance Systems. *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia.

Huang, Y. M., Chen, L. Y. and van Gelder, P. H. A. J. M. (2019). Generalized velocity obstacle algorithm for preventing ship collisions at sea. *Ocean Engineering*, **173**, 142–156.

Huang, Y. M., Chen, L. Y., Negenborn, R. R. and van Gelder, P. H. A. J. M. (2020). A ship collision avoidance system for human-machine cooperation during collision avoidance. *Ocean Engineering*, **217**, 107913.

James, M. K. (1986). Modelling the decision process in computer simulation of ship navigation. *The Journal of Navigation*, **39**, 32–48.

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, **5**(1), 90–98.

Kiss, D. and Tevesz, G. (2012). Advanced Dynamic Window Based Navigation Approach Using Model Predictive Control. *Proceedings of the 17th International Conference on Methods & Models in Automation & Robotics (MMAR)*, Miedzyzdroje, Poland.

Kongsberg Maritime. (2021). *Autonomous Ships*. Available at: https://www.kongsberg.com/maritime/solutions/ship-types/autonomous-ships (Accessed 17 Jan. 2021).

Kreutzmann, A., Wolter, D., Dylla, F. and Lee, J. H. (2013). Towards safe navigation by formalizing navigation rules. *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation*, **7**(2), 161–168.

Kufoalor, D. K. M., Brekke, E. F. and Johansen, T. A. (2018). Proactive Collision Avoidance for ASVS Using A Dynamic Reciprocal Velocity Obstacles Method. *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain.

Kufoalor, D. K. M., Wilthil, E., Hagen, I. B., Brekke, E. F. and Johansen, T. A. (2019). Autonomous COLREGs-Compliant Decision Making Using Maritime Radar Tracking and Model Predictive Control. *Proceedings of the 18th European Control Conference (ECC)*, Naples, Italy.

Kufoalor, D. K. M., Johansen, T. A., Brekke, E. F., Hepsø, A. and Trnka, K. (2020). Autonomous maritime collision avoidance: Field verification of autonomous surface vehicle behavior in challenging scenarios. *Journal of Field Robotics*, **37**, 387–403.

Kuwata, Y., Wolf, M. T., Zarzhitsky, D. and Huntsberger, T. L. (2014). Safe maritime autonomous navigation with COLREGs, using velocity obstacles. *IEEE Journal of Oceanic Engineering*, **39**(1), 110–119.

Larson, J., Bruch, M. and Ebken, J. (2006). Autonomous Navigation and Obstacle Avoidance for Unmanned Surface Vehicles. *Proceedings of SPIE Unmanned Systems Technology VIII, Defense and Security Symposium*, Orlando, USA.

Levander, O. (2017). Autonomous ships on the high seas. *IEEE Spectrum*, **54**(2), 26–31.

https://doi.org/10.1017/S0373463322000315 Published online by Cambridge University Press

Liu, Z. X., Zhang, Y. M., Yu, X. and Yuan, C. (2016). Unmanned surface vehicles: An overview of developments and challenges. *Annual Reviews in Control*, **41**, 71–93.

Loe, Ø. A. G. (2008). Collision avoidance for unmanned surface vehicles. Master's thesis, Norwegian University of Science and Technology. Trondheim, Norway.

Lyu, H. and Yin, Y. (2018). COLREGs-Constrained real-time path planning for autonomous ships using modified artificial potential fields. *The Journal of Navigation*, **72**(3), 1–21.

Martinezgomez, L. and Fraichard, T. (2009). *Collision Avoidance in Dynamic Environments: An* ICS-B*ased* S*olution and its* C*omparative* E*valuation*. *Proceedings of the IEEE International Conference on Robotics & Automation*, Kobe, Japan.

Moe, S. and Pettersen, K. Y. (2017). Set-Based Line-of-Sight (LOS) Path Following with Collision Avoidance for Underactuated Unmanned Surface Vessels Under the Influence of Ocean Currents. *Proceedings of the IEEE Conference on Control Technology & Applications*, Maui, USA.

Open Robotics. (2021). *The Robot Operating System*. Available at: https://www.ros.org/ (Accessed 17 Jan. 2021).

Perera, L. P., Carvalho, J. P. and Soares, C. G. (2011). Fuzzy logic based decision making system for collision avoidance of ocean navigation under critical collision conditions. *Journal of Marine Science & Technology*, **16**(1), 84–99.

Richalet, J., Rault, A., Testud, J. L. and Papon, J. (1978). Model predictive heuristic control: Applications to an industrial process. *Automatica*, **14**, 413–428.

Savkin, A. V. and Wang, C. (2013). A simple biologically inspired algorithm for collision-free navigation of a unicycle-like robot in dynamic environments with moving obstacles. *Robotica*, **31**(6), 993–1001.

Snape, J., Berg, J., Guy, S. J. and Manocha, D. (2011). The hybrid reciprocal velocity obstacle. *IEEE Transactions on Robotics*, **27**(4), 696–706.

Stenersen, T. (2015). *Guidance system for autonomous surface vehicles*. Master thesis, Norwegian University of Science and Technology, Trondheim, Norway.

Wiig, M. S., Pettersen, K. Y. and Krogstad, T. R. (2019). Collision avoidance for underactuated marine vehicles using the constant avoidance angle algorithm. *IEEE Transactions on Control Systems Technology*, **28**(3), 951–966.

Yang, R. (2019). *Research on COLREGs-compliant shipborne autonomous collision avoidance system*. Shanghai, China: Ph.d. thesis, Shanghai Jiao Tong University.

Zheng, Z. (2000). *Research on automatic decision-making system of vessel collision avoidance*. Dalian, China: Ph.d. thesis, Dalian Maritime University.

Zhuang, J., Su, Y., Liao, Y. and Sun, H. (2011). Motion planning of USV based on marine rules. *Procedia Engineering*, **15**, 269–276.

Cite this article: Tao Y and Du J (2022). Agile collision avoidance for unmanned surface vehicles based on collision shielded model prediction control algorithm. *The Journal of Navigation* **75**: 5, 1243–1267. https://doi.org/10.1017/S0373463322000315