Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Collision Avoidance Systems for Maritime Autonomous Surface Ships Considering Uncertainty in Ship Dynamics

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

** Intelligent Transport System Research Center, Wuhan University of Technology, Wuhan, PR China*
***School of Navigation, Wuhan University of Technology, Wuhan, PR China*
(e-mail: yaminhuang, linyingchen@whut.edu.cn)
****Department of Maritime and Transport Technology, Delft University of Technology, Delft, the Netherlands*
*****Safety and Security Science Group, Faculty of TPM, Delft University of Technology, Delft, the Netherlands*
(e-mail: r.r.negenborn,p.h.a.j.m.vangelder@tudelft.nl)

Abstract: Many Collision Avoidance Systems (CAS) for autonomous ships usually presume that a ship's dynamics are completely known in advance. However, precise parameters for ships in different operating conditions are, in fact, uncertain and unknown. The parameter identification of ship dynamics is challenging and time-consuming. Thus, uncertainties in the ship dynamic model are inevitable, which can lead to errors between real trajectories and predicted trajectories. These errors might result in an unexpected collision between ships. Therefore, it is necessary to consider tracking errors in the CAS, which is missing in most existing CAS. This article proposes a way to incorporate the errors in CAS. Specifically, a Velocity Obstacle (VO) algorithm is employed to find collision-free velocities with estimated tracking errors. Firstly, the ship is assumed to be a "black box" whose inputs and outputs are observable, while the internal workings are unknown. Secondly, parameters optimization of a PID controller are employed to determine the best feedback gains for tracking given trajectories; Thirdly, the maximal tracking errors for controlling the ship to arbitrary velocities are estimated. Finally, the maximal error is added to the safety distance and the VO algorithm is employed to find a collision-free solution. The proposed Unknown-Dynamics CAS (UD-CAS) can support the upgrade of existing conventional ships to Type I-III maritime autonomous surface ship.

Keywords: ship control, obstacle avoidance, unknown dynamics, velocity obstacle algorithm.

1. INTRODUCTION

Preventing collisions are crucial for the safety of conventional ships, which motivates the development of various Maritime Autonomous Surface Ships (MASS). Numerous collision avoidance systems (CAS) have been proposed in past decades. Most CAS hold the assumption that ship's dynamics are known and deterministic, i.e., as a white box (Zheng, Negenborn et al. 2017, Huang, Chen et al. 2020). However, in practice, the dynamics of a ship are usually unknown, i.e., as a black box.

To handle the uncertainties in ship dynamics, one reasonable way is identifying ship dynamics beforehand or in real time. A vast of methods has been developed for offline and online parameter identifications. For offline parameter identifications, researchers use experiments in water tank to collect data, build the manoeuvrability model, and regress the parameters (Liu, Hekkenberg et al. 2015). For online parameter identification, researchers usually employed methods, such as Supported Vector Machine, etc., to identify parameters of simplified models (Liu, Zheng et al. 2019). Adaptive control (Haseltalab and Negenborn 2019) and Model Predictive Control (Eriksen, Eriksen et al. 2019) are usually used to control the ship. In general, most CASs presume that the identified dynamics are precise, and decisions are made based on this assumption.

In human-operated ships, navigators can steer the ship to avoid collisions though they have no idea about the formulation of ship dynamics. The experiences of navigators play important roles. In bridge room, the only thing related to the ship dynamics is so-called manoeuvring booklet that shows turning circles, stopping, etc. When a captain/pilot steer a new ship, he checks the booklet and applys different rudders and propellers to get an insight into the performance of the ship, e.g., the response time, tracking errors, etc. Then, he can steer the ship to avoid collisions incorporating the tracking errors. This process inspired us to think about developing a CAS with unknown dynamics, named as Unknown-Dynamics CAS (UD-CAS). The tracking errors are estimated in advance and considered in collision-related decision making.

The structure of this article includes: the description of two basic problems in Section 2; formulations and simplifications of the questions are addressed in Section 3 and 4; simulations are designed to show the performance of the proposed system in Section 5, followed by a discussion section; finally, conclusion is presented in Section 7.

2. PROBLEM STATEMENT

In UD-CAS, instead of identifying ship dynamics, the maximal tracking error is firstly estimated. Then, the error is considered in collision-related decision making.

2.1 Description of Question 1

Since the dynamics are unknow, the Proportional-Integral-Derivative (PID) controller is employed, whose feedback gains are optimized to reduce the tracking error. The optimization problem is noted as Question 1.

The errors of tracking trajectory i (i.e., r_i) is noted as $e^i(t)$. The maximal tracking error over time is noted as e^i_{max}:

$$e^i_{\max} = \max\left(e^i(t)\middle|t \in (0,\infty), r_i, K_{PID}, \chi_0\right), \tag{1}$$

where χ_0 is the given initial state and K_{PID} is feedback gain. Denote e_{max} as the maximal error when the ship tracks all reachable trajectories, i.e.,

$$e_{max} = \max\left(e^i_{\max} \middle| r_i \in \mathrm{RX}, K_{PID}, \chi_0\right), \tag{2}$$

where RX is a set of all reachable trajectories.
To reduce the tracking errors, the feedback grains need to be optimized, which is formulated as follows:

$$e^0_{track} = \min_{K_p,K_I,K_D}\left(e_{\max}\middle|\chi_0\right). \tag{3}$$

The optimal K_{PID} is formulated as:

$$K^*_{PID} = \arg\min_{K_p,K_I,K_D}\left(e_{\max}\middle|\chi_0\right). \tag{4}$$

The tracking errors (i.e., e^0_{track}) are based on a given χ_0. If the initial state is different from χ_0, the tracking errors might be larger than e^0_{track}. Thus, the maximal tracking errors $e^{\max}_{track}$ is calculated as:

$$e^{\max}_{track} = \max\left(e^0_{track}\middle|x_0 \in \chi, K^*_{PID}\right). \tag{5}$$

2.2 Description of Question 2

Question 2 is to incorporate $e^{\max}_{track}$ in collision avoidance. The Velocity Obstacle (VO) algorithm is employed, which helps the vehicle identify the velocities leading to collision.

The foundation of the VO algorithm is the determination of collision condition. A collision happens at time t if and only if the ship is too close to the obstacle, i.e.,

$$P_i(t) \in P_j(t) \oplus ConfP. \tag{6}$$

where, P_i is the position of the ship, P_j is the position of an obstacle, and $ConfP$ is a safety zone. $P_j(t) \oplus ConfP$ refers to a safety zone surrounding the obstacle. The enlarged ConfP considering the max tracking error is formulated as follows:

$$ConfP = \left\{P \middle| \|P - O\|_2 \le \left(R_{safe} + e^{\max}_{track}\right)\right\}. \tag{7}$$

Given the modified collision conditions, a set of velocities leading to violation of $ConfP$ is identified, noted as VO set. In this way, the solution space is cut as collision-free sub-space and dangerous sub-space. Then, the collision avoidance problem turns to be the optimization problem finding an optimal velocity in the collision-free sub-space. Details about the VO algorithm refer to (Alonso-Mora, Naegeli et al. 2015).

3. MAXIMAL TRACKING ERROR

3.1 Formulation of Question 1

To track the desired trajectory (r_i), a PID controller is designed, while feedback gains are uncertain.

Question $1-1$:

$$\begin{aligned}
\min_{K_p,K_I,K_D} \quad & e_{\max} \\
s.t. \quad & Eq.(1) \sim (2), \\
& e^i(t) = r_i(t) - x(t), r_i(t) \in \mathrm{RX} \\
& \mathrm{RX}=\left\{r_x(t)\middle| r_x(t) = \begin{bmatrix}\cos(\psi)\\ \sin(\psi)\end{bmatrix} u\cdot t, v = \begin{bmatrix}u\\ \psi\end{bmatrix} \in RV\right\}, \\
& \dot{x} = f(x,\tau,t), x_0 \in \chi_0, \\
& \tau = K_p e(t) + K_I \int e(s)ds + K_D \dot{e}(\tau),
\end{aligned} \tag{8}$$

where, K_p, K_I, K_D are variables; RV is a set of reachable velocities; the velocity (v) consists of speed u and heading ψ; $f(.)$ is ship dynamics that are unknown, but feasible; τ is the input to the ship system; x_0 is the initial state of the ship consisting of position (x, y), heading, surge speed, sway speed, and yaw rate; χ_0 is a given initial state, defined as:

$$\chi_0 = [0, 0, \pi/2, 0.5, 0, 0]^T$$

For this question, the system is unknown, and it could be nonlinear. Thus, this question is not a convex problem and a global optimal solution is infeasible. Genetic algorithm, then, is employed to find a group of optimal solutions.

3.2 Question Simplification

To save computational time, some simplifications are made. The Reachable Velocity (RV) set is discretized. In principle, RV is an enclosed region that all the velocities that one ship can obtain. In the enclosed space, there are infinite points, which would cost much computational time. Thus, a sampling RV set is adopted, containing 21-by-21 velocity candidates. The velocity candidate consists of the heading and speed. The heading ranges from $\psi_0 - \pi/2$ to $\psi_0 + \pi/2$, where ψ_0 is the heading of the ship. The speed ranges from 0.1 [m/s] to 1 [m/s].

3.3 Error Estimation

Following Eq. (5), the ship with different initial states would be investigated to estimate $e^{\max}_{track}$. To calculate the maximal tracking error with different initial states, the optimal feedback grains in Question 1.1 are used. The ship would pick up one initial state belong to χ ($x_0 \in \chi$), calculate the tracking error with different inputted velocity u_i ($u_i \in \mathrm{RV}$), and find the maximal tracking error w.r.t. a given x_0. This process is repeated until all the initial states in χ are traversed and $e^{\max}_{track}$ is obtained, i.e., Question 1.2:

$$e_{track}^{\max} = \max\left(e_{track}^{0}\left|x_0 \in \chi, K_{PID}^{*}\right.\right),$$
$$\chi = \left[0, 0, \pi/2, u_i, 0, 0\right]^T, \forall u_i \in \text{RV}. \quad (10)$$

4. OPTIMAL EVASIVE ACTION

4.1 Formulation of Question 2

In this part, the maximal error $e_{track}^{\max}$ is added to the safety distance of the ship to guarantee safety.

VO algorithm can find the velocities leading to collision and collects the velocities in a set called VO set. When all the dangerous velocities are collected, the rest of velocities is collision-free. Then, finding a solution from collision-free velocities can be formulated as an optimization problem.

Question 2:

$$\begin{aligned} \min \quad & (v - v_0)^T H (v - v_0) \\ s.t. \quad & v \in V_{free}, \\ & V_{free} = (RV \setminus \text{VO}), \\ & \text{VO} = \bigcup_{t=0}^{\infty} \text{sVO}(t), \\ & \text{sVO}(t) = \frac{1}{t}\left(P_j(t) - P_i(0)\right) \oplus \frac{1}{t} ConfP, \text{ and } Eq.(7). \end{aligned} \quad (11)$$

where, v is the velocity that the PID controller needs to track, and H is a cost matrix. $P_j(0)$ is the position of the ship at the beginning, and $P_j(t)$ is position of the obstacle at time t. VO set is a set of velocities leading to collisions. $RV \setminus \text{VO}$ is a set of reachable collision-free solutions. Since the optimal v has to be safe, v has to belong to V_{free}.

4.2 Assumption and Simplification

The construction of the VO set needs the trajectory of the moving obstacle to be known or predictable. Thus, in this article, we assume that the ships are sharing their trajectories.

Assumption: the trajectory of the obstacle is known by communications or predictions.

In Section 3.2, the RV set is discretizing as a 21-by-21 grid. Thus, this problem becomes a mixed-integer programming problem. As the sampling RV set is small scale, the solution to Question 2 can be solved by checking the reachable velocities. The algorithm is presented as follows.

Algorithm 1: find an optimal collision-free solution from RV set
v_{cand} = RV($v \notin VO$)
if $v_{cand} \neq \varnothing$
J =(v_{cand}-v_0)^{T}H (v_{cand}-v_0)
i = argmin (J)
v^* = $v_{cand}(i)$
return
else
v^* = [0, ψ_0]
end

5. SIMULATION AND RESULTS

5.1 Setup

CyberShip II (CSII) is set as a sample ship, whose dynamics are unknown for the CAS. However, we have a simulator of CS II. The length of the ship is 1.255 [m], noted as L_{ship}. The maximal speed of the scaled ship is set as 1 [m/s], which is roughly 16 knots in real size. The initial velocity of the ship is set as 0.5 [m/s] and the initial heading is set as 090.

For Question 1, genetic algorithm is used, where the boundaries of K_P, K_I, and K_D are set as:

$$lb = 0.01 \times \left[1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1\right]$$
$$ub = \left[100.01 \quad 100.01 \quad 100.01 \quad 10 \quad 10 \quad 10 \quad 100 \quad 100 \quad 100\right]$$

The genetic algorithm developed in Matlab 2019b is employed in this study. By 200 generations, the optimal K_P*, K_I*, and K_D* are found, which are K_P = diag([100.01, 100.01, 100.01]), K_I = diag([10.01, 10.01, 0.01]), and K_D = diag([0.01,0.01,5.7159]).

The tracking process of the ship has been presented in Fig. 1, where the ship starts from its initial state χ_0 and tracks one inputted velocity (1 [m/s], 30 [°]). The optimal K_P^*, K_I^*, and K_D^* are employed. In the last panel, the error between the tracking trajectory and the real trajectory increased dramatically at the beginning, while it was reducing as time goes on. The max error is less than 0.46 [m].

By repeating the above process and recording the max tracking errors, the error map is presented in Fig. 2, where the ship starts from the χ_0 and tracks each velocity in RV set. From this map, it is observed that: when the speed of tracking velocities is fixed, the error goes to minimal as the headings of the tracking velocities close to the initial heading; the more the speed of the tracking velocity close to the initial speed, the smaller the error would be. The max tracking error in this case, i.e., e_{track}^{0}, is less than 0.70 [m]. When the $x_0 \in \chi$, the max tracking error, i.e., $e_{track}^{\max}$, is estimated to be 0.80 [m].

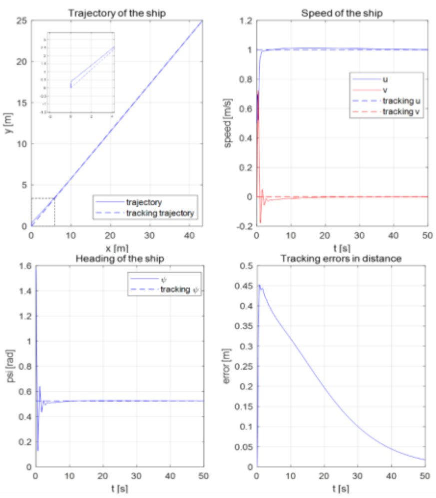


Fig. 1 Outcomes of tracking one velocity from RV set with K_P*, K_I*, and K_D*.

H in Eq. (11) is set as a diagonal matrix: diag([100,100]). Besides, v_0 is the initial speed of the ship, which is 0.5 m/s.

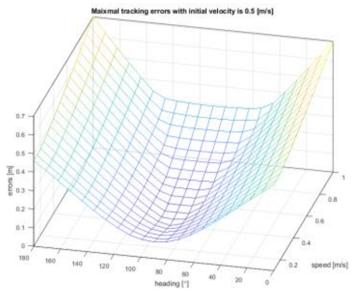


Fig. 2 Max tracking errors when the initial state is χ_0 .

5.2 Scenario: encountering with a dynamic obstacle

In this section, the ship with UD-CAS encounters with the obstacle that is in front of the ship. The obstacle has the speed of (0.5 [m/s]) heading to the South. To test the performance of UD-CAS in extreme conditions, the distance to the obstacle is increased from L_{ship} to 10.5 L_{ship}, i.e., 1.255 to 13.1775[m], see Table 1. The diameter of the obstacle is 1.255 [m]. Thus, the origianl safety distance is 1.255 [m].

Three groups of experiments are designed:

- Standard Group (SG): VO is used in a holonomic vehicle;
- Control Group 1 (CG1): CSII uses VO directly;
- Control Group 2 (CG2): CSII uses UD-CAS.

In SG, the ship is the holonomic vehicle that has perfect manoeuvrability and enables to change speed and course immediately. In CG1 and CG2, the ship with unknown dynamics is used, i.e., CS II. Moreover, the ship is controlled by the PID controller with the optimal feedback gains. Both CGs employ the VO algorithm to avoid collision. Specifically, the ship makes one decision every 2 seconds, and the predicted horizon is 30 seconds. It implies that the velocity resulting in collisions beyond 30 seconds is seen as temporarily collision-free.

Table 1. Results of different groups in Scenario 2

Test Index	L_i [m]	SG	CG1	CG2
1	L_{ship}	√	×	×
2	1.5 L_{ship}	√	×	×
3	2 L_{ship}	√	×	√
4	2.5 L_{ship}	√	×	√
5	3 L_{ship}	√	×	√
6	3.5 L_{ship}	√	×	√
7	4 L_{ship}	√	×	√
8	4.5 L_{ship}	√	√	√
9	5 L_{ship}	√	√	√
10	5.5 L_{ship}	√	×	√
11	6 L_{ship}	√	√	√
…	...	…	…	…
20	10.5 L_{ship}	√	√	√

The difference between these two groups is the size of *ConfP* and the conditions for triggering new decisions. In CG1, the original *ConfP* (radius is 1.255 [m]) is used, while in CG2, the enlarged *ConfP* (radius is 2.05 [m]) is used. Collision avoidance process is triggered whenever the velocity of the ship is in the VO set. However, in CG2, there are two exceptions: the ship's yaw rate is larger than 0.05 rad/s, or the ship violates the buffer area but outside of the original *ConfP*. In these cases, the ship is executing the last action that is collision-free. Thus, no new avoidance process is set off. The outcomes of each test are presented in Table 1.

In Table 1, we find that using VO algorithm for holonomic vehicles can support collision avoidance in these distances even the ship is on the boundary of ConfP (safety zone). In a long distance, all these groups can contribute to successful collision avoidance. However, the ship cannot avoid collision in close ranges, see the results in the CG1 and CG2 columns, when the distance is smaller than 2 L_{ship}. If VO is applied directly, collision avoidance is not guaranteed when the initial distance is smaller than 4.5 L_{ship}, (CG1 column in Table 1). Moreover, the performance of the algorithm is unstable. In some tests, the collision would happen. Generally, when the distance is larger than/equal to 6 L_{ship}, the original VO can help the ship to avoid collision.

UD-CAS, on the other hand, can support the ship to avoid collisions in most distances listed in the table, see the CG2 column in Table 2. In Test 1 and Test 2, the ship is inside the enlarge *ConfP* at the beginning, which is one of exceptions that the system would not trigger new decisions.

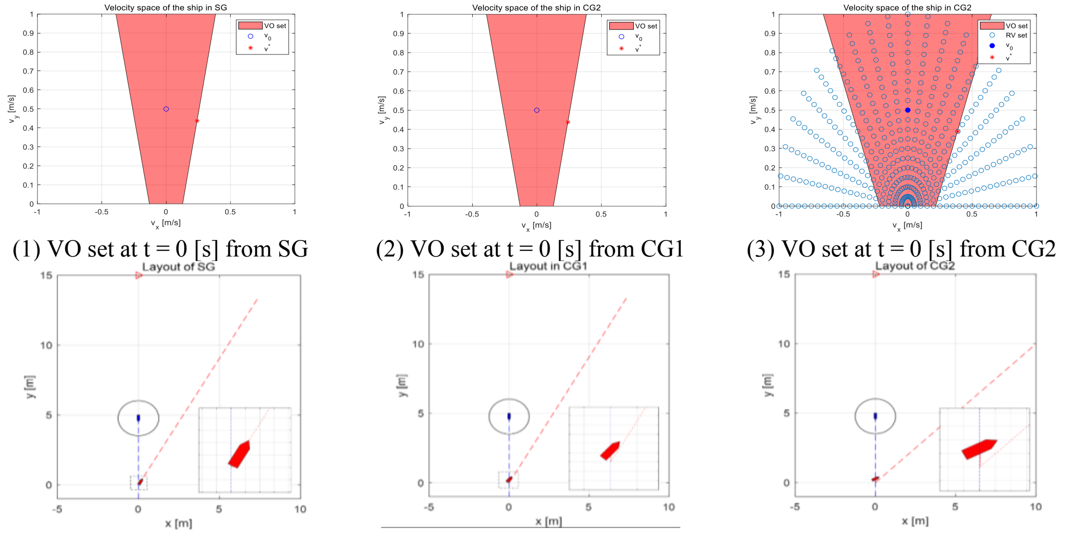
The initial states that lead to inevitable collision given the controller and algorithms is named as "inevitable collision state" (ICS) [12]. The ICS of the ship using settings in CG1 is smaller than 7.53 [m], while ICS using settings in CG2 is less than 2.51 [m]. Therefore, when the relative distance is larger than 6L_{ship}, the ship can follow the suggestion from VO. However, when the relative distance is smaller than 6L_{ship}, the solutions obtained by VO become unreliable. Enlarging the ConfP (safety zone) becomes necessary.

Details of Test 7 are explained to demonstrate the collision avoidance process in SG, CG1, and CG2. The initial distance between ships is 4L_{ship} and other settings are identical to other tests. Fig. 3 shows the initial encountering scenario and Fig.4 shows the entire trajectories in different groups.

At 0 [s], the velocity spaces of the CSII in three groups are presented in Fig. 3 (1) to (3). Panel (1)-(2) are identical at this moment, which show the VO sets of the ship in SG and CG1. Panel (3) shows VO set in CG2 and the light blue "○" are all reachable velocity, i.e., RV set. In these panels, the blue "o" represents the initial velocity of the CSII, which are all in the VO sets. It implies, the ship is in danger and evasive action is needed. The red "*" is the optimal collision-free solution. The solutions from different groups all suggest the ship to turn right-hand side. However, CG2 requires the ship to take more turning operations since the *ConfP* is enlarged.

At 0.5 [s], the collision-free solutions found at 0 [s] influence the motions of the CSII. Details are shown in Fig. 3 (4) to (6). In SG, due to perfect manoeuvrability, the ship tracks the inputted velocity and corresponding trajectory (dotted red line) immediately, see panel (4). In CG1 and CG2, the ship changes its heading, while the inertia drives the ship move forward departing from the tracking trajectory.

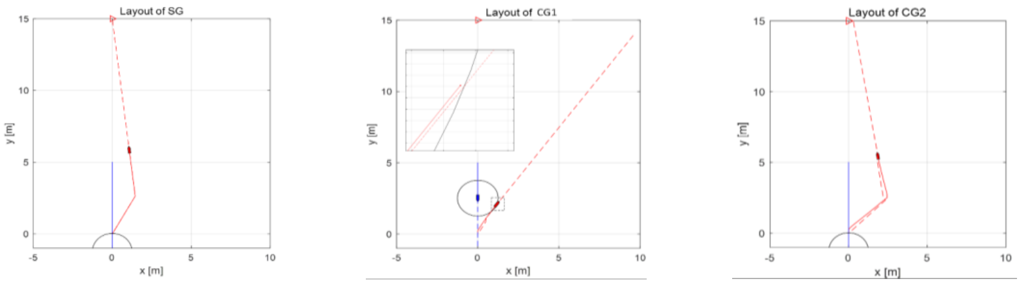
The trajectories are shown in Fig. 4. In panel (1), the trajectories in SG are relatively sharp. In CG1, two ships collided at 4.5 [s], see panel (2), and the small window shows the violation of *ConfP*. The dotted red line is the expected trajectory to avoid the obstacle. However, the ship has not converged to the expected trajectory and there is no buffer zone for tracking errors. In CG2, the tracking errors are considered as the additional buffer zone to the *ConfP*. Although the ship does not reach the expected trajectory at each time slice, the tracking errors are converging, and are considered in UD-CAS.



(1) VO set at t = 0 [s] from SG (2) VO set at t = 0 [s] from CG1 (3) VO set at t = 0 [s] from CG2

(4) Layout at t = 0.5 [s] from SG (5) Layout at t = 0.5 [s] from CG1 (6) Layout at t = 0.5 [s]from CG2

Fig. 3 VO sets and Layouts at the beginning of simulation in SG, CG1, and CG2



(1) SG at 12 [s] (2) CG1 at 4.5 [s] (3) CG2 at 12 [s]

Fig. 4 Trajectories of ships in SG, CG1, and CG2

6. DISCUSSION

In Section 5, a series of simulations are designed to show the performance of the proposed UD-CAS. In this section, the reasons for the failures in Section 5 are discussed, together with potentials of the proposed UD-CAS.

6.1 Discussion on failures of the original VO algorithm

In Section 5, as the initial distance reduced, the original VO would lead to collisions. In fact, the failures of using original VO algorithm in close ranges are foreseeable since the precondition of using VO algorithm is that the vehicle should be holonomic, while the ship is not.

VO algorithm collects a set of velocities at present (say time t_0) that leading to collision. Then, the identified solution is collision-free if the ship changes its motion immediately at current position and at time t_0. The decision is valid only at t_0. Since ships have inertia, the execution of the decision needs time and space. Suppose that the ship reaches the "collision-free velocity" at t_1. Then, in the updated scenario, the collision-free solution at time t_0 might not still collision-free at t_1. In brief, the decision making and execution are not synchronous, see Fig. 5(1). Subsequently, a new collision-avoidance solution is needed at t_1 and the previous processes repeat. In return, the ship would take a series of small evasive actions to avoid the collision, which is not encouraged by navigational regulations, e.g., COLREGs.

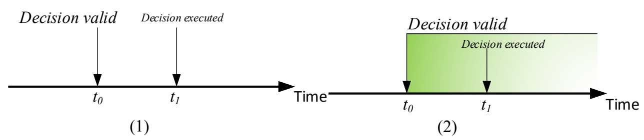


Fig. 5 Decision valid time and execution time in (1) original VO algorithm and (2) UD-CAS

UD-CAS incorporating the control errors solves this problem by extending the valid time of decision. Specifically, decision making at time t_0 has considered the tracking errors in the future. Suppose the ship is holonomic, then, the distance to the obstacle will always be larger than safety distance adding $e_{track}^{\max}$, i.e.,

$$\left\|\bar{d}_{ij}(t)\right\|_2 \geq R_{safety} + e_{track}^{\max}. \qquad (12)$$

However, ship is non-holonomic vehicle. Thus, the real time distance to the obstacle is bounded as:

$$\left\|\bar{d}_{ij}\left(t\right)\right\|_2 + e_{track}^{\max} \geq \left\|d_{ij}\left(t\right)\right\|_2 \geq \left\|\bar{d}_{ij}\left(t\right)\right\|_2 - e_{track}^{\max} . \quad (13)$$

Combining Eq. (12) and (13), we have

$$\left\|d_{ij}\left(t\right)\right\|_2 \geq R_{safety} . \quad (14)$$

Therefore, the decision made by UD-CAS at t_0 has a longer valid time than that in the original VO algorithm, see Fig. 5 (2), which solves the non-synchronous problem. Besides, an additional side effect of using UD-CAS on board is avoiding a series of small operations.

6.2 Potentials in the transition towards MASS

In recent years, the development of MASS has attracted numerous attentions. Researchers focused on developing fully autonomous ships that is, i.e., MASS Type IV. Model ships with complete knowledge about the dynamics is used.

For the existing ships, the dynamics are difficult to be estimated, which is one main challenge of upgrading the conventional ships towards MASS. The UD-CAS offers a new tool for this problem.

When the dynamics of ships are formulated but contain highly non-linear or high-order terms, many CAS linearize the system or ignore the high-order terms, e.g. GVO-CAS developed in (Huang, Chen et al. 2019). For instance, researchers have spent vast efforts in identifying parameters of a scaled model ship called KVLCC2 to improve the forecast of the ship's manoeuvrability. However, the system of KVLCC2 has high-order terms, and the inputs and outputs have non-linear relationships. Few CAS are suitable for these types of system, while UD-CAS can offer supports.

In practical situations, the dynamic of ships are unknown. The optimization using Genetic algorithm and find the maximal controller errors become challenging. To use the proposed UD-CAS, some modifications are needed. One solution is that using the known or arbitrary feedback gains and measure the maximal tracking errors; another solution is asking human operators of the ship to input the maximal tracking error; or, replacing the PID controller by adaptive controllers that are developed to reduce the tracking errors when the ship dynamics are unknown. As long as the tracking error is bounded, the UD-CAS can work properly. However, when the maximal tracking error increases, the usage of the ship in restrict waters might be influenced.

5. CONCLUSION

In this article, the collision avoidance system handling unknown ship dynamics is developed, named as Uncertainty Integrated CAS (UD-CAS). To handle the unknown dynamics, two mains steps are introduced. Firstly, we estimate the tracking errors and find the maximal error: a PID controller is designed to track inputted trajectories; then, feedback gains of the PID controller are optimized to reduce the tracking errors. Secondly, the maximal tracking error is incorporated in collision avoidance. Velocity Obstacle (VO) algorithm is employed to find an optimal collision-free solution incorporating the enlarged safety zone.

Simulation experiments are carried out to demonstrate the proposed system. Specifically, series of heading scenarios are designed to compare the performance of using original VO algorithm and the proposed UD-CAS on board ship. The dynamic of the ship is assumed to be unknown but accessible, just like a black box. The results show that when the ship encounters a dynamic obstacle, the original VO algorithm and UD-CAS can both support collision avoidance when the obstacles are at far distances. However, as the initial relative distance reduced, the original VO algorithm becomes unreliable and it would lead to collision, while the UD-CAS can still support the ship to avoid collision.

ACKNOWLEDGEMENTS

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

REFERENCES

Ali Haseltalab and R. R. Negenborn (2019). "Adaptive control for autonomous ships with uncertain model and unknown propeller dynamics." Control Engineering Practice **91**: 104116.

Alonso-Mora, J., T. Naegeli, R. Siegwart and P. Beardsley (2015). "Collision avoidance for aerial vehicles in multi-agent scenarios." Autonomous Robots **39**(1): 101-121.

Eriksen B. H., M. Breivik, Wilthil E. F., A. L. Flåten, E. Brekke (2019). The Branching-Course MPC Algorithm for Maritime Collision Avoidanc. Journal of Field Robotics, 2019: 1222-1249.

Huang, Y., L. Chen, P. Chen, R. R. Negenborn and P. H. A. J. M. van Gelder (2020). "Ship collision avoidance methods: State-of-the-art." Safety Science **121**: 451-473.

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

Liu, C., H. Zheng, R. R. Negenborn, X. Chu and S. Xie (2019). "Adaptive predictive path following control based on least squares support vector machines for underactuated autonomous vessels." Asian Journal of Control: 1-17.

Liu, J., R. Hekkenberg, E. Rotteveel and H. Hopman (2015). "Literature review on evaluation and prediction methods of inland vessel manoeuvrability." Ocean Engineering **106**: 458-471.

Zheng, H. R., R. R. Negenborn and G. Lodewijks (2017). "Fast ADMM for Distributed Model Predictive Control of Cooperative Waterborne AGVs." IEEE Transactions on Control Systems Technology **25**(4): 1406-1413.