Parallel distributed collision avoidance with intention consensus based on ADMM

Hoang Anh Tran * **Tor Arne Johansen** *
Rudy R. Negenborn **

* *Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Norway (e-mail: hoang.a.tran@ntnu.no, tor.arne.johansen@ntnu.no).*
** *Department of Maritime and Transport Technology, Delft University of Technology, The Netherlands (e-mail: r.r.negenborn@tudelft.nl)*

Abstract: This paper presents an approach to the problem of collaborative collision avoidance of autonomous inland ships. We propose a distributed model predictive control algorithm that allows ships to negotiate their intention to collaboratively avoid collisions directly. Furthermore, a new method is introduced to better shape ships' behavior in order to follow traffic regulations. The simulation results illustrate that the proposed algorithm could significantly increase ship safety navigation. Moreover, it also shows that the solution proposed by our algorithm complies with waterway traffic regulations except in complex situations when other safe solutions are negotiated.

Keywords: Marine Transportation; Modeling, Control and Optimization of Transportation Systems; Connected and Automated Vehicles; ADMM; Model predictive control.

---

1. INTRODUCTION

Ships sailing in inland waterway traffic usually must cooperate to avoid collisions. Developing a collaborative collision avoidance system (C-CAS) becomes one of the challenges when it comes to autonomous inland ships. This paper presents a distributed approach to solve the problem of C-CAS for ships in inland waterways.

In the recent decade, the development of intention-sharing concepts (Guiking, 2022; STM, 2015) has not only allowed ships to share intentions but also opened the opportunity for intention negotiation between autonomous ships. Several approaches have been proposed to allow ships to cooperatively avoid collision (Ferranti et al., 2018; Chen et al., 2018, 2020). Two main frameworks widely used for the C-CAS are the centralized and distributed framework. As in (Chen et al., 2018), the centralized framework uses a coordinator to determine a collision-free path for all ships within a region. On the contrary, in a distributed framework, each ship solves its collision avoidance problem with concern for neighboring ships' intentions. In comparison, a distributed framework is more robust regarding communication instability than a centralized one and offers better network scalability, i.e., increasing the number of participant ships (Akdağ et al., 2022).

When developing a C-CAS, it is important that the algorithm complies with the waterway traffic regulations. A common approach to overcome this challenge is using a binary variable to represent a ship's compliance status (violation or not) (Johansen et al., 2016). An alternative solution used in MPC-based algorithms is introduced as a potential function (Eriksen et al., 2020). However, only a few of existing C-CAS algorithms for inland autonomous ships explicitly consider inland traffic regulations (Tran et al., 2024).

In this research, we adopt the Alternating Method of Multipliers (ADMM) to develop our distributed Model Predictive Control (MPC) algorithm for the problem of C-CAS. Specifically, the Nonlinear ADMM (NADMM) algorithm, proposed by (Themelis and Patrinos, 2020), is used to solve our nonlinear MPC problem. Our main contributions are twofold. Firstly, we propose a C-CAS algorithm that allows ships to directly negotiate their intention with neighboring ships to reduce potential collision risks. Different from (Tran et al., 2024), where a ship decides only its future trajectory, in our proposed algorithm, a ship influences other ships by proposing the future trajectory for itself and others. Furthermore, the C-CAS algorithm proposed in this paper is performed in parallel between ships instead of serially, as in (Tran et al., 2024). Secondly, we introduce a method that makes the behavior of ships more consistent with inland waterway traffic regulations. Moreover, the performance of the proposed C-CAS is verified in several representative scenarios using simulation experiments.

The rest of this paper is organized as follows. Section 2 introduces the overall structure of the control system, assumptions, and traffic regulations that are considered. The proposed distributed C-CAS algorithm is described in

---

⋆ The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 955.768 (MSCA-ETN AUTOBarge), and the Researchlab Autonomous Shipping at Delft University of Technology. This publication reflects only the authors' view, exempting the European Union from any liability. Project website: http://etn-autobarge.eu/.

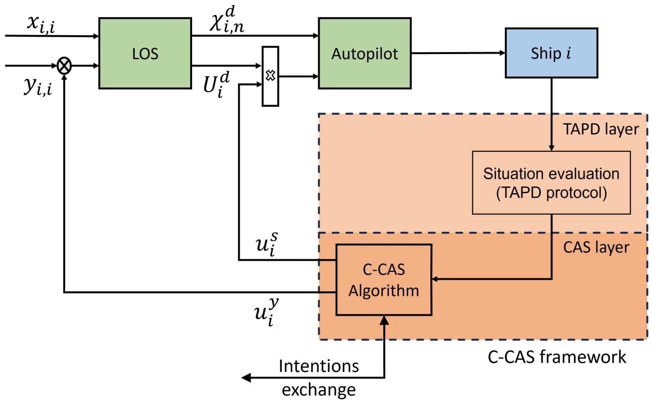


Fig. 1. Control scheme with the proposed C-CAS framework: U_i^d and $\chi^d_{i,n}$ are desired thrust and course angle in the inertial coordinate frame $\{n\}$; the control signals from CAS are cross-track offset, u_i^y, and speed modification, u_i^s

.

Section 3. Section 4 presents the evaluation of the proposed algorithm through several simulation experiments. Finally, conclusions and future research are given in Section 5.

2. PRELIMINARIES

This research considers the problem of C-CAS within a set $\mathcal{M}$ of M ships. The problem is solved in a distributed manner. Each ship solves its local C-CAS problem and exchanges the solution with neighboring ships. Through negotiation and communication, a local optimal consensus solution is reached.

We use the two-layer framework that was proposed by (Tran et al., 2024). The first layer is the traffic assessment & priority determination (TAPD) protocol, where ships determine give-way or stand-on priority with other neighboring ships based on traffic regulations. The second layer is the C-CAS algorithm, which decides the collision avoidance action for the ship considering traffic regulations. The overall control scheme is presented in Fig. 1. The ship is guided from waypoint to waypoint by line-of-sight (LOS) guidance. When a potential risk appears, e.g., dynamic obstacles, the C-CAS shall be activated to modify the trajectory that the LOS guidance provides to avoid collision. This paper focuses on the C-CAS algorithm in the second layer. We propose a distributed MPC approach that provides collaborative collision avoidance solutions in accordance with the TAPD protocol.

2.1 Traffic assessment and priority determination protocol

According to (Tran et al., 2024), the TAPD is established with the following two steps:

(1) Depending on the situation with other ships and following the traffic rules, every ship i ($i \in \mathcal{M}$) assigns a relative priority value for all surrounding ships j, including itself. A ship with a lower priority value has to give-way to a ship with higher one.
(2) All ships compare their priority values pair-wise to identify their priority with neighboring ships.

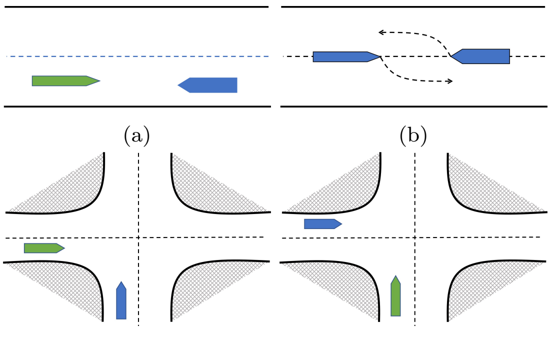


Fig. 2. Traffic situations according to the Netherlands' inland shipping police regulations: (a), (b) head-on situations; (c), (d) crossing situations. Blue ship is the give-way ship, and green is stand-on ship.

In this research, we apply the traffic regulations adopted in Chapter 6 of the Netherlands' inland shipping police regulations (BPR, 2017) to determine the give-way or stand-on priorities. The following rules are considered:

- Head-on situation: If two vessels are approaching each other on opposite courses in such a way that there is a risk of collision, the vessel not following the starboard side of the fairway shall give-way to the vessel following the starboard side of the fairway (see Fig. 2a). If neither vessel follows the starboard side of the fairway, each shall give-way to vessels on the starboard side so that they pass each other port to port (see Fig. 2b).
- Crossing situation: If the courses of two ships cross each other in such a way that there is a risk of collision, the vessel not following the starboard side of the fairway shall give-way to the vessel following the starboard side of the fairway (see Fig. 2c). In case none of the ships follows the starboard side of the fairway, the ship approaching from the port side gives way to the vessel approaching from starboard (see Fig. 2d).
- Overtaking situation: A vessel overtaking another vessel should keep out of the way of the overtaken vessel.

2.2 Assumptions

We assume that every ship in set $\mathcal{M}$ has sufficient computation and communication equipment to execute the C-CAS algorithm and broadcast the information to all other ships in $\mathcal{M}$. We also assume that the communication delay is sufficient small, and that the communication is secure and reliable.

3. PARALLEL DISTRIBUTED COLLABORATIVE COLLISION AVOIDANCE ALGORITHM

This section proposes a distributed MPC algorithm to solve the collision avoidance problem for inland autonomous ships. The C-CAS algorithm adds to the LOS guidance a cross-track offset (u_i^y) and speed modification (u_i^s) to adjust the trajectory to avoid potential collisions.

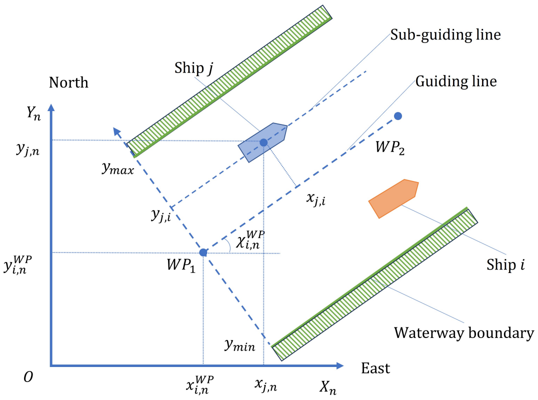


Fig. 3. Path coordinate and inertial coordinate.

3.1 *Kinematic model of ships*

Let us introduce the path coordinate frame $\{\varpi_i\}$ as in (Zheng et al., 2016) to define the position of an arbitrary ship j, i.e., ship with index $j \in \mathcal{M}$, along guiding lines between the defined waypoints of ship i. The position of ship j with respect to the path coordinate frame $\{\varpi_i\}$ is denoted as $p_{j,i} = [x_{j,i}, y_{j,i}, \chi_{j,n}]^\top$ The transformation from the inertial coordinate frame $\{n\}$ to the path coordinate frame $\{\varpi_i\}$ is as follows:

$$p_{j,i} = \begin{bmatrix} x_{j,i} \\ y_{j,i} \\ \chi_{j,i} \end{bmatrix} = R_i \begin{bmatrix} x_{j,n} - x_{j,n}^{WP} \\ y_{j,n} - y_{i,n}^{WP} \\ \chi_{j,n} - \chi_{i,n}^{WP} \end{bmatrix} = R_i(\eta_j - \eta_i^{WP}),$$

$$R_i = \begin{bmatrix} \cos\left(\chi_{i,n}^{WP}\right) & \sin\left(\chi_{i,n}^{WP}\right) & 0 \\ -\sin\left(\chi_{i,n}^{WP}\right) & \cos\left(\chi_{i,n}^{WP}\right) & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where $\eta_j = [x_{j,n}, y_{j,n}, \chi_{j,n}]^\top$ is the state of ship j with respect to inertial frame $\{n\}$, and the parameters of the previous active waypoint in the inertial frame $\{n\}$ is $\eta_i^{WP} = [x_{i,n}^{WP}, y_{i,n}^{WP}, \chi_{i,n}^{WP}]^\top$ (see Fig. 3).

Following (Tran et al., 2023), we have the kinematic model of ship i with respect to coordinate frame $\{\varpi_i\}$ as follows:

$$\begin{aligned} x_{i,i}(k+1) &= x_{i,i}(k) + u_i^s(k) U_i^d \cos(\chi_{i,i}(k))\Delta T, \\ y_{i,i}(k+1) &= y_{i,i}(k) + u_i^s(k) U_i^d \sin(\chi_{i,i}(k))\Delta T, \\ \chi_{i,i}(k+1) &= \chi_{i,i}(k) \\ &\quad + \frac{\Delta T}{T_1}\big[\chi_i^{max} \tanh(K_e(u_i^y(k) - y_{i,i}(k))) \\ &\qquad - \chi_{i,i}(k)\big], \end{aligned} \tag{1}$$

with u_i^y, u_i^s being the cross-track offset and speed modification, respectively. Moreover, the nominal surge speed of ship i is U_i^d, the maximum steering angle that ship i can achieve in a sampling period ΔT is χ_i^{max}; K_e is a tuning parameter of the LOS guidance; and T_1 is a positive constant depending on the characteristics of the autopilot and ship hydrodynamics. Additionally, we denote the control vector of ship i as $u_i(k) = [u_i^y(k), u_i^s(k))]^\top$.

3.2 *Collision risk evaluation*

Algorithm 1 Parallel collaborative collision avoidance

```
Input: $s = 0$, establish $W_i$
while  $s \leq s_{\max}$ do
    for all $i \in \mathcal{M}$ in parallel do
        receive $\hat{\xi}_j^s$ from neighboring ships
        update the global variable $\xi^{s+1}$ follows (6a)
        $\bar{p}_i^{s+1} = \tilde{R}_i(\xi - \bar{\eta}_i^{WP})$
        update local variables follows equations (6b)–(6e)

        $\hat{\xi}_j^s = \tilde{R}_i^{-1}\hat{p}_i^s + \bar{\eta}_i^{WP}$
        Transmit data $\hat{\xi}_i^s$ to all ship $j \in \mathcal{M}_i$.
        $s = s + 1$.
    end for
end while
```

We introduce the risk function, $R_{ij}(t_0+k)$, to predict the collision risk of ship i with respect to ship j at k times steps from the present time t_0. Accordingly, the value of $R_{ij}(t_0+k)$ increases as the distance between ship i and j is reduced, and is approximately zero if this distance is large enough. The C-CAS algorithm can increase the safety of the ship's navigation by minimizing the risk function. The risk function is selected as follows:

$$R_{ij}(t_0+k) = \frac{K_{ca}}{\sqrt{1+K_d k}} D_x(t_0+k) D_y(t_0+k),$$

where K_{ca} is predefined constant based on safety criteria that depends on the traffic situation. A discount factor $\frac{1}{\sqrt{1+K_d k}}$, with $K_d \geq 0$, is used to reduce the weight of the collision risk as k increases because a collision prediction further away from t_0 is less critical and accurate than that early on. Besides, $D_x(t_0+k)$, $D_y(t_0+k)$ are defined as follows:

$$D_x(t_0+k) = exp\left[-\frac{(x_{i,i}(t_0+k) - x_{j,i}(t_0+k))^2}{\alpha_{xj}}\right],$$

$$D_y(t_0+k) = exp\left[-\frac{(y_{i,i}(t_0+k) - y_{j,i}(t_0+k))^2}{\alpha_{yj}}\right],$$

with α_{xj} and α_{yj} being parameters linked to the size and shape of ship j.

3.3 *Problem formulation*

Let us introduce $\tilde{p}_{i,i}(t_0)$ and $\tilde{u}_i(t_0)$ as vectors containing the system state and input over a control horizon of N time steps, i.e., $\tilde{u}_i(t_0) = [u_i^\top(t_0), u_i^\top(t_0+1), ..., u_i^\top(t_0+N-1)]^\top$, $\tilde{p}_{i,i}(t_0) = [p_{i,i}^\top(t_0), p_{i,i}^\top(t_0+1), ..., p_{i,i}^\top(t_0+N)]^\top$. For every ship $i \in \mathcal{M}$, we define $\tilde{p}_i = [\tilde{p}_{1,i}, \tilde{p}_{2,i}, ..., \tilde{p}_{M,i}]^\top$ as a local state variable of all ships in $\mathcal{M}$ with respect to the coordinate frame $\{\varpi_i\}$. Additionally, we denote the global variable $\xi = [\xi_1, \xi_2, ..., \xi_M]^\top$ to record the consensus solution between ships, where ξ_i is the consensus trajectory of ship i over the horizon of the MPC. The following constraint must be satisfied to guarantee the consensus of intention between ships:

$$\tilde{p}_i = \tilde{R}_i(\xi - \bar{\eta}_i^{WP}), \quad \forall i \in \mathcal{M}, \tag{2}$$

where $\bar{\eta}_i^{WP} = \mathbf{1}_{N+1} \otimes \eta_i^{WP}$, with $\otimes$ denoting the Kronecker product, and $\tilde{R}_i$ defined as follows:

$$\tilde{R}_i = \begin{bmatrix} R_i & & 0 \\ & \ddots & \\ 0 & & R_i \end{bmatrix} \in \mathbb{R}^{3(N+1)M \times 3(N+1)M}.$$

The cost function of ship i for the MPC is then formulated as follows:

$$\mathcal{J}_i(\tilde{p}_i, \tilde{u}_i) = \mathcal{J}_i^{ca}(\tilde{p}_i) + \mathcal{J}_i^{e}(\tilde{u}_i) + \mathcal{J}_i^{b}(\tilde{u}_i). \tag{3}$$

where $\mathcal{J}_i^{ca}(\tilde{p}_i)$ is the sum of risk functions with all neighboring ships over the horizon, i.e., $\mathcal{J}_i^{ca}(\tilde{p}_{i,i}) = \sum_{k=1}^{N+1} \sum_{j \in \mathcal{M} \setminus \{i\}} R_{ij}(t_0 + k)$. $\mathcal{J}_i^{e}(\tilde{u}_i)$ is the cost of control actions, emphasizing that ship i should alter its course only when it can substantially reduce the risk of collision. $\mathcal{J}_i^{e}(\tilde{u}_i)$ is defined as follows:

$$\begin{aligned} \mathcal{J}_i^{e}(\tilde{u}_i) = \sum_{k=1}^{N+1} \Big[& K_y \left(u_i^y(t_0 + k) - u_i^y(t_0 + k - 1) \right)^2 \\ & + K_s \left(1 - u_i^s(t_0 + k) \right)^2 \Big], \end{aligned} \tag{4}$$

where K_y, K_s are positive control parameters. Additionally, $\mathcal{J}_i^{b}(\tilde{u}_i)$ is a term that makes the behavior of ship i adequately represent waterway traffic regulations, e.g., steering towards starboard in a head-on situation. More details on $\mathcal{J}_i^{e}(\tilde{u}_i)$ and $\mathcal{J}_i^{b}(\tilde{u}_i)$ can be found in (Tran et al., 2024).

Remark 1. The decision variable $\tilde{p}_i$ of $\mathcal{J}_i^{ca}(\tilde{p}_i)$ contains the state variables of all ships in $\mathcal{M}$. Therefore, each ship, besides deciding its own trajectory, also proposes the trajectory for other neighboring ships.

We formulate the distributed MPC collision avoidance problem of ship $i \in \mathcal{M}$, with cost function (3) as follows:

$$\min_{\tilde{p}_i, \tilde{u}_i} \quad \mathcal{J}_i(\tilde{p}_i, \tilde{u}_i) \tag{5a}$$

$$\text{s.t.:} \quad p_{i,i}(t_0 + k + 1) = f_i(p_{i,i}(t_0 + k), u_i(t_0 + k)), \tag{5b}$$

$$p_{i,i}(t_0) = p_{i,i}^{init}, \tag{5c}$$

$$u_i(t_0 + k) \in U_i, \tag{5d}$$

$$W_i \tilde{p}_i = W_i \tilde{R}_i (\xi - \bar{\eta}_i^{WP}), \tag{5e}$$

where f_i represents the kinematics of ship i defned by (1), and the boundary set of the control input is U_i. Moreover, $W_i = \mathbf{diag}(w_{1i}, w_{2i}, ..., w_{Mi}) \otimes I_{3N+1}$ is a weighted matrix that will be discussed later in this section.

We define the feasible state/input region for ship i as $\mathbf{G}_i := \left\{ [\tilde{u}_i^\top, \tilde{p}_{i,i}^\top]^\top | \text{(5b), (5c), (5d) are satisfied} \right\}$. Moreover, we denote $\bar{p}_i^{s+1} = \tilde{R}_i(\xi - \bar{\eta}_i^{WP})$ being the global variable that is transform to the path coordinate frame $\{\varpi_i\}$. Then, following to (Themelis and Patrinos, 2020), we have the NADMM update for the controller of ship i at iteration index s as follows:

$$\xi^{s+1} = \frac{1}{M} \sum_{j=1}^{M} \hat{\xi}_j^s, \tag{6a}$$

$$z_i^{s+1/2} = z_i^s + \beta(1 - \lambda) \left(\tilde{p}_i^s - \bar{p}_i^{s+1} \right), \tag{6b}$$

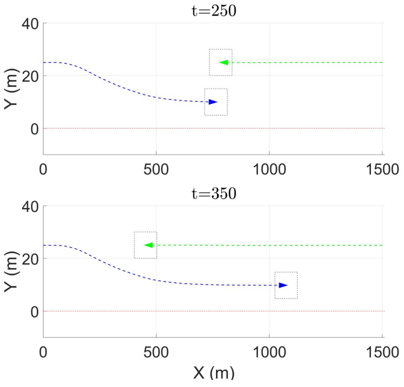
$$\begin{bmatrix} \tilde{u}_i^{s+1} \\ \tilde{p}_i^{s+1} \end{bmatrix} = \operatorname{argmin}_{[\tilde{u}_i, \tilde{p}_{i,i}]^\top \in \mathbf{G}_i} \left\{ \mathcal{L}_i(\tilde{p}_i, \tilde{u}_i) \right\}, \tag{6c}$$

$$z_i^{s+1} = z_i^{s+1/2} + \beta \left(\tilde{p}_{i,i}^{s+1} - \bar{p}_{i,i}^{s} \right), \tag{6d}$$

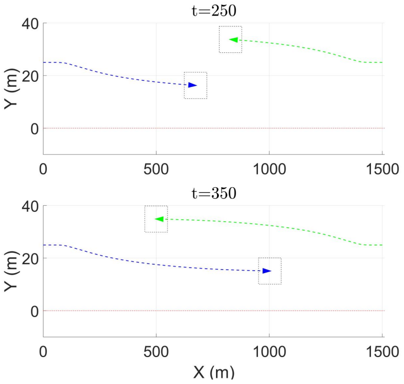
$$\hat{p}_i^{s+1} = \tilde{p}_i^{s+1} + \frac{1}{\beta} z_i^{s+1}, \tag{6e}$$

$$\hat{\xi}_j^{s+1} = \tilde{R}_i^{-1} \hat{p}_i^{s+1} + \bar{\eta}_i^{WP}, \tag{6f}$$

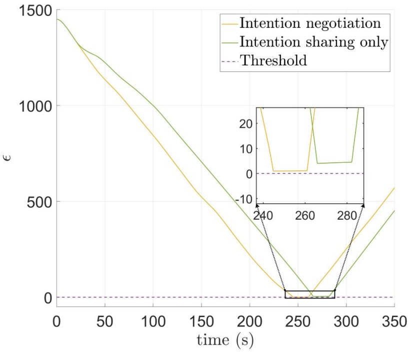
where $\mathcal{L}_i(\tilde{p}_i, \tilde{u}_i) = \mathcal{J}_i(\tilde{p}_i, \tilde{u}_i) + \left\langle z_i^{s+1/2}, W_i \left(\tilde{p}_i - \bar{p}_i^{s+1} \right) \right\rangle + \frac{\beta}{2} \| W_i (\tilde{p}_i - \bar{p}_i^{s+1}) \|^2$ with z_i being the Lagrange multiplier. The detailed steps of the parallel collaborative collision avoidance algorithm are presented in Algorithm 1. In contrast to the original NADMM update in (Themelis and Patrinos, 2020), the update of the global variable ξ is divided into three steps (6e), (6f) and (6a). This allows the update to be performed in parallel without the need for a coordinator to update the global variable. Furthermore, instead of sending two variables, i.e., z_i^{s+1} and $\tilde{p}_i^{s+1}$, each ship only has to send one variable, i.e., $\hat{p}_i^{s+1}$.



(a) Intention sharing and intention negotiation.



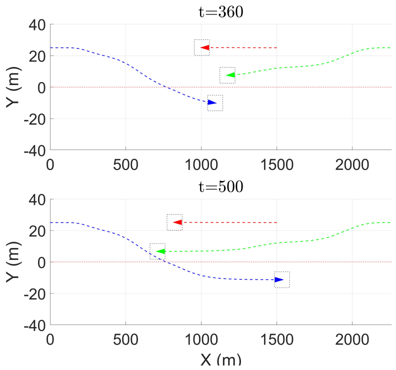
(b) Intention sharing only.



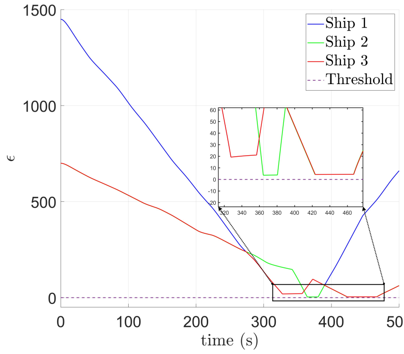
(c) Safety index ϵ_1.

Fig. 4. Case 1.1: Head-on situation. Ship 1, and ship 2 are illustrated in blue, and green triangles, respectively.

From the update of $\tilde{p}_i^{s+1}$ in (6c), we observe that if we set w_{ji} large enough, then the solution for the trajectory of ship j proposed by ship i, $\tilde{p}_{j,i}$, would be approximately equal to that of the global variable, i.e., $\tilde{p}_{j,i} \approx \xi_j$. Consequently, the decision of ship i is to change trajectory to avoid collision (if it exist) with ship j. On the other hand, if w_{ji} is small enough, ship i will keep its current trajectory and request ship j to change trajectory (by



(a)



(b) Safety index ϵ_1.

Fig. 5. Case 1.2: Head-on with overtaking situation. Ship 1, ship 2, and ship 3 are illustrated in blue, green, and red triangles, respectively.

modifying $\tilde{p}_{j,i}$). Based on this observation, we establish the weight matrix W_i as follows:

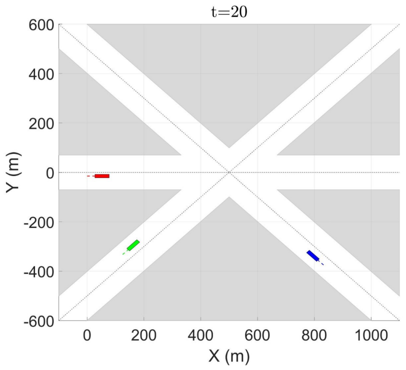
- $w_{ii} = 1$.
- If ship i can stand on to ship j, then $w_{ji} = K_{SO}$, with $K_{SO} > 0$ is small enough.
- If neither of ship i nor j has stand on priority over each other and $\upsilon_i \geq \upsilon_j$, then $w_{ji} = \alpha_\upsilon K_{GW}$, with $K_{GW} > 0$ is large enough, and $\alpha_\upsilon \in (0, 1)$.
- $w_{ji} = K_{GW}$, otherwise.
- All other elements of W_i are zeros.

In which υ_i is a secondary parameter, e.g., weight or length of ship. We use the secondary parameter to give a slightly higher priority to a larger ship in case none of both ships has stand on priority over each other. The stand-on or give-way priority between ship i and j is determined using TAPD protocol.

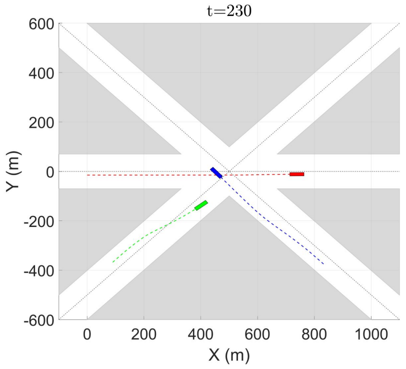
Remark 2. We distinguish the NADMM update (6) from that presented in (Tran et al., 2024) by the way ships negotiate on their intention. In (Tran et al., 2024), ships negotiate their intention indirectly through a decision order, and the NADMM update is done serially. In contrast, the NADMM update (6) allows ships to directly influence each other's solution through (6e) and (6a). Additionally, instead of relying on a decision order, the update (6) is performed in parallel, and the weight matrix W_i shapes the actions of ships to follow traffic regulations.

4. SIMULATION EXPERIMENTS

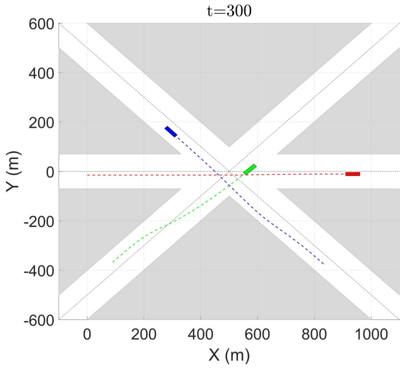
In this section, we perform several simulation experiments to evaluate the performance of the proposed C-CAS algorithm. The proposed algorithm is evaluated in two cases:



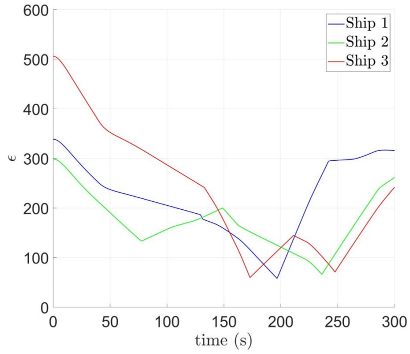
(a)



(b)



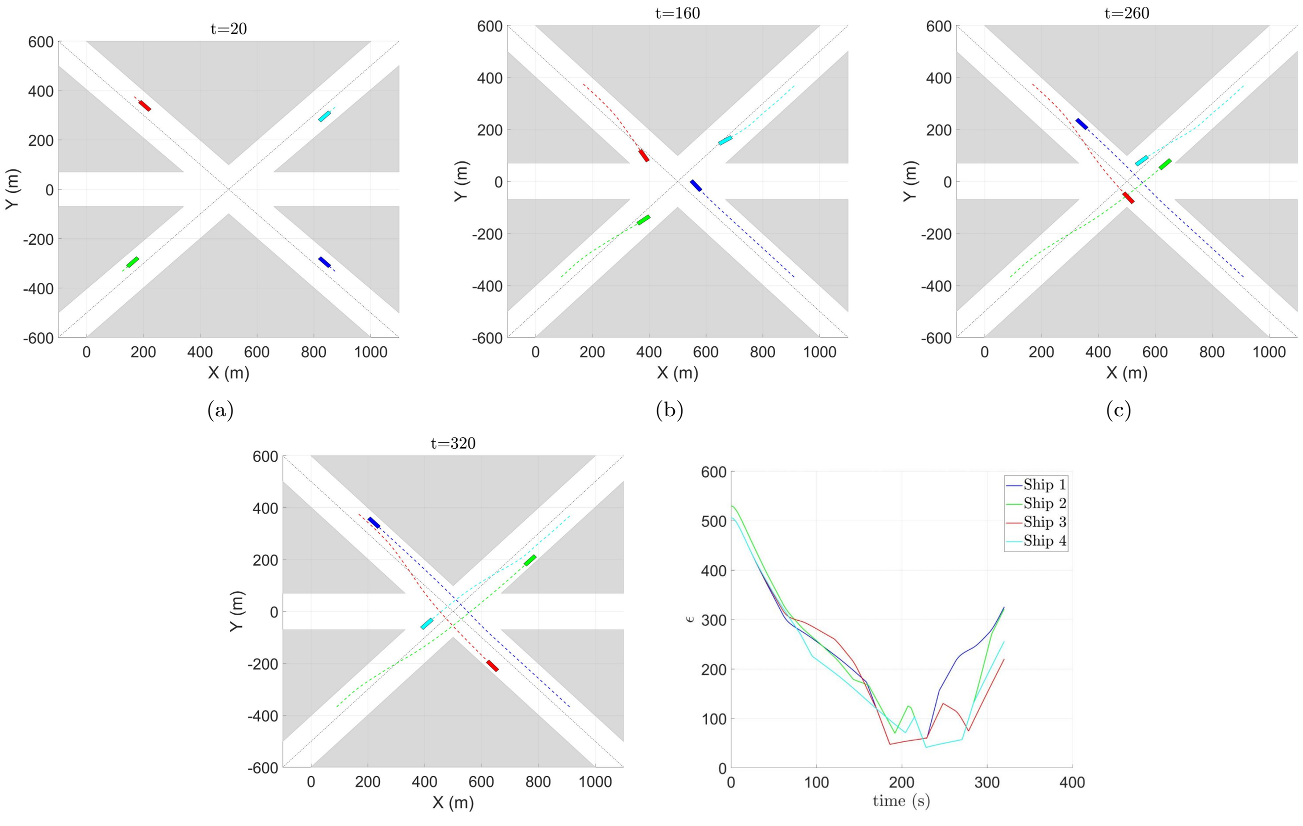
(c)



(d) Safety index of each ship.

Fig. 6. Case 1.3: Intersection crossing between 3 ships. Ship 1, ship 2, and ship 3 are illustrated in blue, green, and red rectangles, respectively.

(1) Case 1 - Simple scenarios: Ships encounter head-on, overtaking, or crossing situations.



(a) (b) (c)

(d) (e) Safety index of each ship.

Fig. 7. Case 2.1: Intersection crossing between 4 ships, scenario 1. Ship 1, ship 2, ship 3, and ship 4 are illustrated in blue, green, red, and cyan rectangles, respectively.

(2) Case 2 - Complex scenarios: More than two ships encounter a situation that involves both head-on and crossing situations.

We define the safety index of ship i as follows:

$$\epsilon_i = \min_{j\in\mathcal{M}, j\neq i} \{\max\{|x_{i,i} - x_{j,i}| - d_i^x, |y_{i,i} - y_{j,i}| - d_i^y\}\},$$

where d_i^x, d_i^y are the safety distances of ship i in x and y axes of the coordinate frame υ_i. If $\epsilon_i \leq 0$, then there is a ship j is in the safety zone of ship i, and a collision is likely to happen. The goal of the C-CAS algorithm is to guarantee $\epsilon_i > 0$.

The control parameters are chosen as: $K_y = 10^{-2}$, $K_s = 2 \times 10^{-2}$, $\beta = 3 \times 10^{-4}$, $K_{SO} = 0.5$, and $K_{GW} = 10^3$. All ships have the same safety distance: $d^x = 50$, $d^x = 14$. The optimization problem (6c) is solved by the Casadi toolbox (Andersson et al., 2019) using the interior point optimizer (IPOPT).

4.1 Head-on, overtaking, or crossing scenarios

In simple scenarios, ships encounter each other in either a head-on, overtaking or crossing scenario. The stand-on and give-way priorities in the case are easy to determine, and ships are expected to follow traffic regulations strictly.

Fig. 4 shows the head-on scenario in which ship 1 sails on the port side must give way to ship 2, which sails on the starboard side of the waterway. The proposed algorithm provides a good collision avoidance solution that complies with the considered traffic regulations (see Fig. 4a and 4c). On the contrary, when two ships only share intention without negotiation, i.e., the ADMM scheme and weight matrix W_ij are not used, as shown in Fig. 4b, both ships make a starboard turn to avoid collision. Furthermore, we can see from Fig. 4c that two ships made a larger and unnecessary turn to avoid collision in case of only sharing intention.

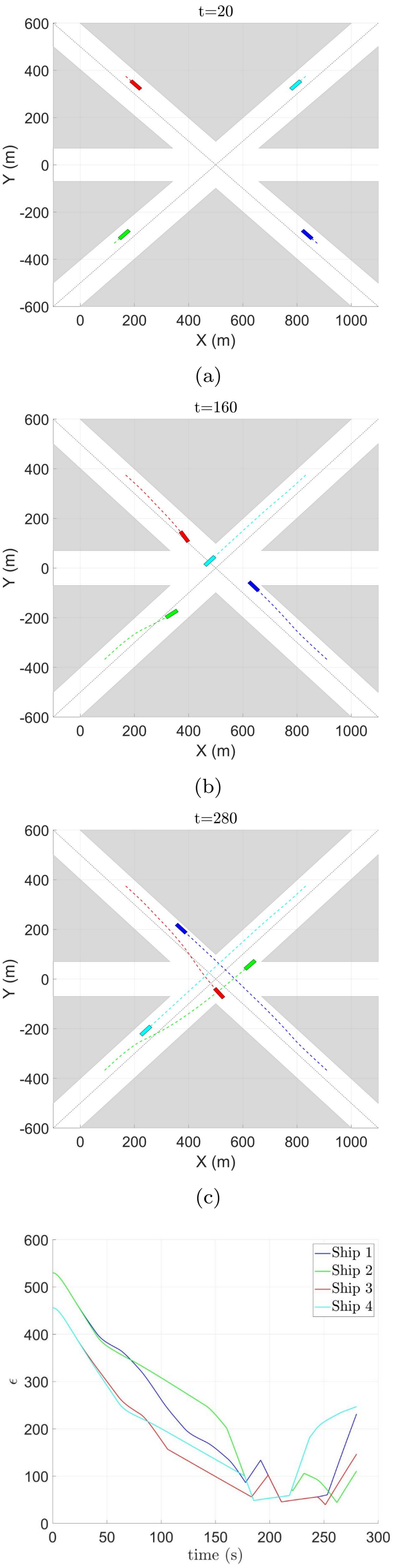
The scenario in Fig. 5 involves three ships, where ship 3 is sailing at a lower speed than ship 2. Since ship 2 is overtaking ship 3, ship 2 must give way to ship 3. The results (as in Fig. 5a) show that all ships follow the traffic rules.

A crossing situation involving three ships is shown in Fig. 6. The expected priority in this scenario is for ship 3 to stand on, ship 1 to give way to ship 3, and ship 2 to give way to both ship 1 and 3. Results in Fig. 6b and 6c illustrate that the proposed C-CAS complies with the considered traffic rules.

4.2 Combined head-on and crossing scenarios

Next, we evaluate the performance of the proposed algorithm in complex scenarios where ships are simultaneously involved in both head-on and crossing situations.

Fig. 7a shows a scenario of four ships crossing an intersection, and ships 1 and 3 are in a head-on situation. As shown in Fig. 7b, ship 1 is the first to cross the intersection because it sails on the starboard side of the waterway. Ship 3 changes course to give way for ship 1 and reduces speed



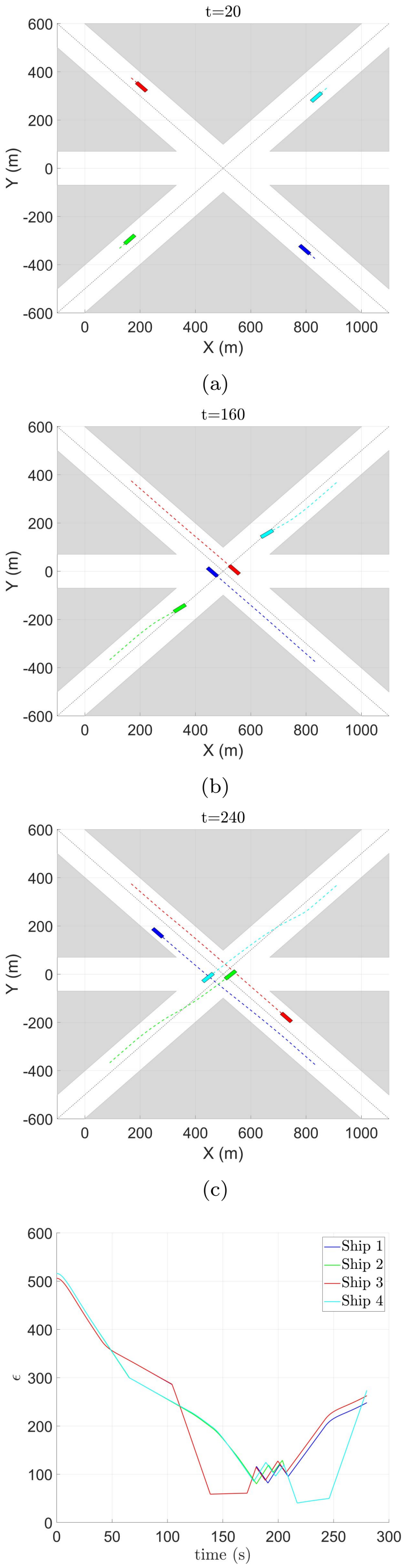
(a)

(b)

(c)

(d) Safety index of each ship.

Fig. 8. Case 2.2: Intersection crossing between 4 ships, scenario 2. Ship 1, ship 2, ship 3, and ship 4 are illustrated in blue, green, red, and cyan rectangles, respectively.



(a)

(b)

(c)

(d) Safety index of each ship.

Fig. 9. Case 2.3: Intersection crossing between 4 ships, scenario 3. Ship 1, ship 2, ship 3, and ship 4 are illustrated in blue, green, red, and cyan rectangles, respectively.

to give way to ship 2 (see Fig. 7c). Ship 4 is the last ship to cross the intersection since it has the lowest priority (see Fig. 7d). Compared to the results in (Tran et al., 2024), with the same situation, the C-CAS in (Tran et al., 2024) failed to comply with the traffic rules. The main difference lies in the introduction of W_{ij} that makes give-way and stand-on priorities become hard constraints and better shape the behavior of ships follow traffic rules.

Fig. 8a shows a similar scenario, however in this situation, ship 1 and 4 have the same priority, and do not have to give way to each other. The results after the negotiation process are shown in Fig. 8b, where ship 1 steers to starboard to give way to ship 4.

Fig. 9a shows a scenario in which each ship must give way to one ship (from its starboard side) and can stand on to another ship (from its port side). The solution by the proposed algorithm is for ships 2 and 4 to give way to ships 1 and 3, as shown in Fig.9b - 9c. There is a deviation from traffic regulations due to the complex situation. However, all ships can safely cross the intersection.

5. CONCLUSION AND FUTURE RESEARCH

In this paper, we proposed a distributed MPC that utilizes an ADMM scheme to solve the problem of collaborative collision avoidance for autonomous ships in inland waterways. A parallel distributed MPC algorithm for the C-CAS problem is presented that allows ships to exchange and negotiate their intentions directly. Through iterative intention sharing and negotiating under the ADMM scheme, ships can reach a consensus on intention. Furthermore, a weight matrix is used to guide the behavior of ships in complying with traffic regulations. In typical simulation traffic scenarios, the proposed C-CAS algorithm helps ships avoid collision and comply with traffic regulations simultaneously.

Future research will focus on improving the robustness of the C-CAS algorithm in the presence of lossy communications. Field experiments will also be considered to verify the algorithm's performance in practical conditions.

REFERENCES

Akdağ, M., Solnør, P., and Johansen, T.A. (2022). Collaborative collision avoidance for maritime autonomous surface ships: A review. *Ocean Engineering*, 250, 110920.

Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., and Diehl, M. (2019). CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1), 1–36. doi:10.1007/s12532-018-0139-4.

BPR (2017). Binnenvaartpolitiereglement. URL `https://wetten.overheid.nl/BWBR0003628/2017-01-01`.

Chen, L., Huang, Y., Zheng, H., Hopman, H., and Negenborn, R.R. (2020). Cooperative multi-vessel systems in urban waterway networks. *IEEE Transactions on Intelligent Transportation Systems*, 21(8), 3294–3307. doi:10.1109/Tits.2019.2925536.

Chen, L., Negenborn, R.R., and Hopman, H. (2018). Intersection crossing of cooperative multi-vessel systems. *IFAC Papersonline*, 51(9), 379–385. doi:10.1016/j.ifacol.2018.07.062.

Eriksen, B.O.H., Bitar, G., Breivik, M., and Lekkas, A.M. (2020). Hybrid collision avoidance for ASVs compliant with COLREGs Rules 8 and 13-17. *Frontiers in Robotics and AI*, 7. doi:10.3389/frobt.2020.00011.

Ferranti, L., Negenborn, R.R., Keviczky, T., and Alonso-Mora, J. (2018). Coordination of multiple vessels via distributed nonlinear model predictive control. In *2018 European Control Conference (ECC)*, 2523–2528. IEEE, Limassol, Cyprus.

Guiking, C. (2022). Digital intention sharing : simulation study on the benefits of intention sharing. URL `https://open.rijkswaterstaat.nl/overige-publicaties/2022/digital-intention-sharing-simulation/`.

Johansen, T.A., Perez, T., and Cristofaro, A. (2016). Ship collision avoidance and colregs compliance using simulation-based control behavior selection with predictive hazard assessment. *IEEE Transactions on Intelligent Transportation Systems*, 17(12), 3407–3422. doi:10.1109/Tits.2016.2551780.

STM (2015). Voyage exchange format and architecture, MONALISA 2 D1.3.2. STM MONALISA. URL `https://stm-stmvalidation.s3.eu-west-1.amazonaws.com/uploads/20160420144429/ML2-D1.3.2-Voyage-Exchange-Format-RTZ.pdf`.

Themelis, A. and Patrinos, P. (2020). Douglas-Rachford splitting and ADMM for nonconvex optimization: tight convergence results. *SIAM Journal on Optimization*, 30(1), 149–181. doi:10.1137/18m1163993.

Tran, H.A., Johansen, T.A., and Negenborn, R.R. (2023). A collision avoidance algorithm with intention prediction for inland waterways ships. *IFAC-PapersOnLine*, 56(2), 4337–4343. doi:10.1016/j.ifacol.2023.10.1805.

Tran, H.A., Johansen, T.A., and Negenborn, R.R. (2024). Distributed MPC for autonomous ships on inland waterways with collaborative collision avoidance. URL `http://arxiv.org/abs/2403.00554`.

Zheng, H., Negenborn, R.R., and Lodewijks, G. (2016). Predictive path following with arrival time awareness for waterborne AGVs. *Transportation Research Part C-Emerging Technologies*, 70, 214–237. doi:10.1016/j.trc.2015.11.004.