Distributed model predictive control for vessel train formations of cooperative multi-vessel systems

Chen, Linying; Hopman, Hans; Negenborn, Rudy R.

DOI
10.1016/j.trc.2018.04.013

Publication date
2018

Document Version
Final published version

Published in
Transportation Research Part C: Emerging Technologies

Transportation Research Part C 92 (2018) 101–118

Contents lists available at ScienceDirect

Transportation Research Part C

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

Distributed model predictive control for vessel train formations of cooperative multi-vessel systems

Linying Chen*, Hans Hopman, Rudy R. Negenborn

Department of Maritime and Transport Technology, Delft University of Technology, Delft, The Netherlands

ARTICLE INFO

Keywords:
Cooperative multi-vessel systems
Vessel train formation
Distributed model predictive control
Autonomous vessels
ADMM

ABSTRACT

Recently, the cooperative control of multiple vessels has been gaining increasing attention because of the potential robustness, reliability and efficiency of multi-agent systems. In this paper, we propose the concept of Cooperative Multi-Vessel Systems (CMVSs) consisting of multiple coordinated autonomous vessels. We in particular focus on the so-called Vessel Train Formation (VTF) problem. The VTF problem considers not only cooperative collision avoidance, but also grouping of vessels. An MPC-based approach is proposed for addressing the VTF problem. A centralized and a distributed formulation based on the Alternating Direction of Multipliers Method (ADMM) are investigated. The distributed formulation adopts a single-layer serial iterative architecture, which gains the benefits of reduced communication requirements and robustness against failures. The impacts of information updating sequences and responsibility parameters are discussed. We furthermore analyze the scalability of the proposed method. Simulation experiments of a CMVS navigating from different terminals in the Port of Rotterdam to inland waterways are carried out to illustrate the effectiveness of our method. The proposed method successfully steers the vessels from different origins to form a vessel train. Due to the effective communication, vessels can timely respond to the velocity changes that others make. After the formation is formed, the distances between vessels become constant. The results show the potential to use CMVSs for inland shipping with enhanced safety.

1. Introduction

1.1. Background

Autonomous vessels have been developed for more than 20 years. Many works have been done to improve the autonomy of vessels (see Campbell et al., 2012; Liu et al., 2016). Seeing the advantages of multi-agent cooperation, including robustness, reliability and efficiency, an increasing number of researchers pay attention to the cooperative control of multiple vessels in recent years.

The advantages of cooperation between vessels are also the reasons that motivate the related research. Firstly, cooperation can enhance the safety of waterborne transport with communication between vessels. In current waterborne transport systems, vessels do not actively coordinate their actions with others. When encountering other vessels, vessels may misunderstand the intentions of others, which may lead to oscillation (Van den Berg et al., 2008) and even collisions (Kujala et al., 2009). Communication among the vessels can provide additional information, such as data about the objects beyond the reach of sensors, the intentions of others, etc. The additional information can assist agents in negotiating and collaborating with others to take effective actions. Secondly, inland

* Corresponding author.
E-mail addresses: L.chen-2@tudelft.nl, chenlinying@outlook.com (L. Chen).

https://doi.org/10.1016/j.trc.2018.04.013
Received 30 June 2017; Received in revised form 4 April 2018; Accepted 16 April 2018
Available online 07 May 2018
0968-090X/

shipping can benefit from Vessel-to-Vessel and Vessel-to-Infrastructure cooperation. For instance, with cooperation, vessels can coordinate their voyage plans to avoid congestions at ports and locks (Li et al., 2016). Furthermore, when combining voyage planning with infrastructure scheduling, vessels can adjust their speed to arrive at a required time, and make better use of infrastructure resources (Li et al., 2015). Thirdly, with cooperation, a group of vessels can carry out tasks more efficiently and effectively, such as search and rescue, ocean sampling, hydrographic survey etc. (see Liu et al., 2016). Applications such as towing of large structures, underway replenishment and tandem loading need cooperation, as well.

Therefore, in this paper, we focus on the cooperation of multiple vessels. We propose the concept of a Cooperative Multi-Vessel System (CMVS). A CMVS is a system in which vessels utilize Vessel-to-Vessel (V2V) and Vessel-to-Infrastructure (V2I) communication to negotiate and collaborate with each other for the aim of improving overall safety, efficiency, or for performing specific tasks. As the first step, we hereby in particular focus on the V2V cooperation in a CMVS.

1.2. Related works

A number of methods have been proposed for the cooperative control of multi-agent systems, see Olfati-Saber et al. (2007), Murray (2007) and Ren and Beard (2008) and the references therein. Regarding the similarity of vessels and vehicles, existing studies on cooperative driving of vehicles can provide valuable references for the study of vessel coordination. Nevertheless, those methods and algorithms cannot directly be applied to the control of vessels. So far, the main focus of cooperative driving has been on longitudinal control (Monteil et al., 2014; Hafner et al., 2013). However, due to the large inertia and hydrodynamic influences, the execution time needed to accelerate or decelerate a vessel is much longer than to accelerate or decelerate a vehicle. In practice, steering (lateral control) is regarded as the ordinary practice of seamen to avoid collisions. Moreover, the movement of vessels is significantly affected by the external environment, such as wind, wave, and current, which brings more uncertainties in vessel motion control.

Recently, some approaches have been proposed for the cooperative control of vessels. Depending on the goals of cooperation, two types of research are found in the literature: formation tracking and collision avoidance. Formation tracking aims at steering a group of vessels to form a specific geometric configuration and move along a given path. Learning from the formation control of vehicles, most studies on vessel formation tracking employ the three approaches (Ihle, 2006), i.e., leader-follower architecture (Almeida et al., 2010; Shojaei, 2015; Liu and Bucknall, 2015), behavioral methods (Arrichiello et al., 2006), and virtual structures (Ihle et al., 2006), while considering the characteristics of vessels and external disturbances.

In the research on cooperative collision avoidance, vessels only communicate and cooperate with others when there are collision risks. In existing non-cooperative collision avoidance methods, such as potential field (Daily and Bevly, 2008), velocity obstacles (Huang et al., 2018), and optimization-based methods (Zhang et al., 2015), vessels have to predict the actions that other vessels may take. Instead, in the methods for cooperative collision avoidance, vessels share their intentions. The actions of the involved vessels are determined by following a certain protocol (Tam and Bucknall, 2013) or negotiating through iterations (Zheng et al., 2016). Tam and Bucknall (2013) proposed a cooperative path planning method for collision avoidance using the regulation, COLREGS, as a protocol. In Zheng et al. (2016, 2017), a coordinator was responsible for the coupling collision avoidance constraints. Agreements among vessels were reached through iterations alternate between the coordinator and local path following controllers.

However, the cooperative behavior of vessels in a CMVS for transporting goods is neither typical formation tracking nor cooperative collision avoidance. When sailing in ports, waterways, or canals, it is not necessary for vessels to maintain a specific configuration. Nevertheless, collision avoidance is not the only connection between vessels. For instance, vessels can share voyage plans to avoid a long waiting time at ports or locks; sailing in groups also help to keep the vessels being connected, especially when we consider the effective range of ship-borne sensors, which help them to deal with unexpected changes; another attractive advantage to motivate vessels sailing in groups is the potential of reduced drag forces (Ihle et al., 2006).

Due to the limitation of waterway boundaries, the cooperative sailing of vessels will result in train-liked formations. Therefore, we define the cooperative behavior of vessels in a CMVS as moving into a Vessel Train Formation (VTF). This behavior relaxes the constraints on formation keeping. At the same time, compared with cooperative collision avoidance, vessels in a VTF enjoy the benefits of sailing in groups with a closer connection.

1.3. Contributions

This paper focuses on Vessel Train Formation of Cooperative Multi-vessel Systems. A method based on Distributed Model Predictive Control (DMPC) is proposed to solve the VTF problem. The proposed method has a single-layer distributed structure. Each vessel determines its own actions. Agreements are achieved via serial iterative negotiations. Through communication, the formation and safety constraints are satisfied, and both local and overall performance can be improved. Compared with existing approaches, the proposed method has the following characteristics.

Firstly, the proposed method solves the VTF problem with a single-layer control structure. Existing cooperative methods usually consider multiple layers, in which a leader or coordinator is used to deal with coupling constraints. For example, in Zheng et al. (2016), there is a coordinator responsible for the collision avoidance constraint. Iterations are alternating between the coordinator and each vessel until agreements have been reached. On the contrary, there is no coordinator in our framework. The final agreement is achieved when the trajectory that a vessel finds is the same as the one it sends to others in the former iteration. The updating of variables is carried out independently for each vessel. This means that at each iteration, information only needs to be exchanged once.

Secondly, the single-layer distributed framework provides a vessel the freedom to join or leave a CMVS. It also makes the proposed algorithm robust against failures of an individual vessel. For instance, when a vessel fails to find a solution, it can keep the solution found at the former iteration. Other vessels will try to find collision-free solutions with respect to that earlier solution.

Thirdly, the proposed method adopts a serial iteration scheme. As only one vessel performs computations at a time, vessels can have the most up-to-date information from their neighbors. Compared with the method with a parallel iteration scheme as proposed in Zheng et al. (2016), fewer iterations are needed to reach agreements. Moreover, using a serial scheme avoids the non-convergence problem that the parallel scheme without a coordinator may have.

Last but not the least, the information proposed to be exchanged consists of the predictive trajectories over a prediction horizon. The dynamics of the vessels therefore need not necessarily be the same. The proposed algorithm can therefore also be used in setting with heterogeneous vessels.

With the proposed method, the CMVSs can be used for transportation in inland waterways or canal networks. For example, a CMVS consisting of small vessels can replace the work of a large vessel with following advantages. Firstly, small vessels have lower requirements on waterway dimensions than large vessels. Thus, using fleets can greatly improve the accessibility of waterborne transport networks. Secondly, small vessels have more alternative routes when congestions occur. Consequently, using small vessels helps to relieve the pressure on locks, and to enhance the robustness of waterway networks. Thirdly, goods on large vessels for inland shipping usually have ports of call. This may lead to the problems of inefficiency and low utilization rate. Alternatively, applying fleets can evade these issues with more flexible schedules.

1.4. Outline

The remainder of this paper is organized as follows. In Section 2, the dynamics of a CMVS are modeled using the concepts from graph theory. The VTF problem is stated accordingly. Subsequently, a centralized MPC method is proposed to solve the VTF problem in Section 3. In Section 4, a serial iterative DMPC algorithm is proposed based on the Alternating Direction Method of Multipliers (ADMM). The impacts of different factors, such as information exchanging sequences, responsibility parameters, and the scalability of the proposed method, are analyzed in Section 5, followed by a simulation experiment of a CMVS traveling from the Port of Rotterdam to inland waterways. Finally, the conclusions and future research directions are given in Section 6.

2. Vessel train formation of CMVSs

In this section, the framework of a CMVS is provided, and characteristics of the Vessel Train Formation (VTF) are given.

2.1. Modeling of CMVSs

A CMVS consists of multiple autonomous vessels. As shown in Fig. 1, each vessel is controlled by an agent. The agent controls the motion of a vessel at the individual layer, while cooperating with other agents at the coordination layer. V2V communication and cooperation help to keep distances between vessels. Through the network level communication, vessels arrive at a lock or a bridge at

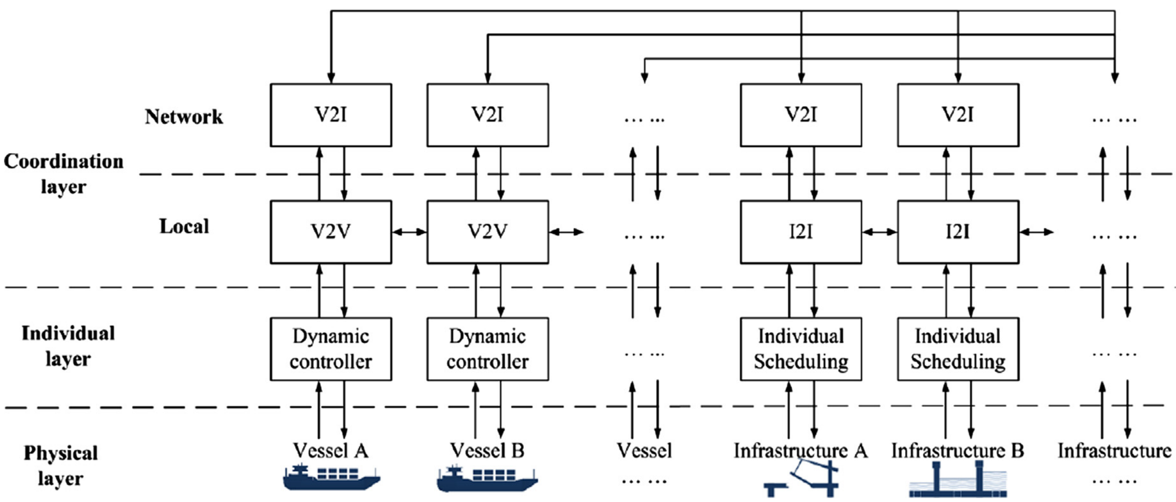

Fig. 1. Architecture of a CMVS.

a predetermined time rather than waiting. In return, infrastructure agents can adjust operation schedules based on the information provided by vessels.

Let the vessels be regarded as mass points. The dynamics of vessel i are given by the following linear discrete-time model:

$$p_i(k+1) = p_i(k) + q_i(k) \tag{1}$$

$$q_i(k+1) = q_i(k) + u_i(k) \tag{2}$$

where p_i, q_i, $u_i \in \mathbb{R}$ denote the position, velocity and acceleration of vessels i, respectively. Therefore, a state space representation for the dynamics is

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k) \tag{3}$$

$$y_i(k) = C_i x_i(k) + D_i u_i(k) \tag{4}$$

where $x_i(k)$ and $y_i(k)$ are the state and output of vessel i at time step k, $x_i(k) = \begin{bmatrix} p_i(k) \\ q_i(k) \end{bmatrix}$; $u_i(k)$ is the input; and $A_i \in \mathbb{R}^{n_{i,x} \times n_{i,x}}$, $B_i \in \mathbb{R}^{n_{i,x} \times n_{i,u}}$, $C_i \in \mathbb{R}^{n_{i,y} \times n_{i,x}}$, $D_i \in \mathbb{R}^{n_{i,y} \times n_{i,u}}$ are the system matrices: $A_i = \begin{bmatrix} I & I \\ 0 & I \end{bmatrix}$, $B_i = \begin{bmatrix} 0 \\ I \end{bmatrix}$, $C_i = [I \quad 0]$, $D_i = 0$.

The dynamics of the vessels are limited by physical limitations, such as maximum speeds, maximum engine power, etc. Hence,

$$u_{\min} \leqslant \|u(k)\|_2 \leqslant u_{\max}, \tag{5}$$

$$q_{\min} \leqslant \|q(k)\|_2 \leqslant q_{\max}, \tag{6}$$

where $\|\cdot\|_2$ is the Euclidean norm.

Due to the limitations of sensors, vessels can only receive and broadcast information over a limited range. Thus, given an interaction range $r_i > 0$, a vessel only communicates and interacts with vessels in this range (i.e., its neighbors). The set of the neighbors of vessel i is

$$N_i = \{j \in \nu : \|p_j - p_i\|_2 \leqslant r_i\}. \tag{7}$$

Then, a CMVS can be represented by a graph $G = (\nu, e)$ that consists of a set of vertices and edges. Vessels are the vertices $\nu = \{1, 2, \cdots, n\}$. The set of edges e represents the communication and interaction possibilities between the vessels:

$$e = \{(i, j) : i, j \in \nu, \|p_i - p_j\|_2 \leqslant \min(r_i, r_j), j \neq i\}. \tag{8}$$

For safe navigation, vessel i should keep a safety distance (d_{safe}) with its neighbors, i.e.,

$$\|p_i - p_j\|_p \geqslant d_{\text{safe}}. \tag{9}$$

where $\|\cdot\|_p$ represents the p-norm.

The positions of vessels change over time. Thus, the CMVS graph is time varying. We use $G(k)$ and $N_i(k)$ to denote the CMVS graph and the relevant parameter at time step k.

2.2. Vessel train formation

In waterway networks, navigable waters are usually limited by the banks, which makes waterways as corridors. Moreover, when navigating, vessels should follow some rules and regulations. For the sake of safety, these rules specify the types of maneuvers that should be taken in some situations. For example, most of the inland waterways regulations suggest the vessels sailing along the starboard side (Rijkswaterstaat and Binnenvaartpolitiereglement, 1983). Therefore, the cooperative sailing of vessels in inland waterways will result in train-liked formations. The train-liked formations are flexible. Vessels can change their positions in the fleet.

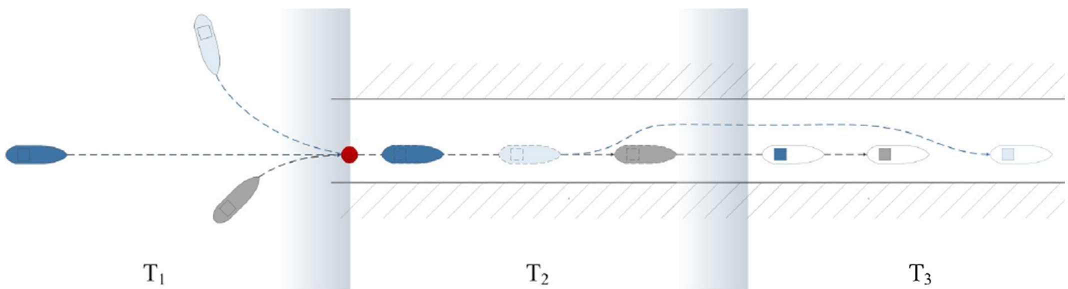

Fig. 2. Vessel train formation.

Thus, lateral operation, i.e., overtaking, is allowed. For example, in Fig. 2, the three vessels decided to form a vessel train at T_1; after the primary vessel train forming at T_2, the second vessel starts overtaking; the new vessels train is formed at T_3. We therefore name the cooperative behavior of vessels in waterways as moving into a *Vessel Train Formation*.

The main function of vessels is to transport goods from one place to another. Therefore, vessels usually have predetermined origins, destinations and paths. In order to exchange information and enjoy the benefits of sailing together, vessels in a CMVS attempt to stay close to each other. At the same time, vessels should not collide with others. Thus, in the VTF problem, following three rules are applied:

(1) Path following: attempt to follow the predetermined paths;
(2) Aggregation: attempt to stay close to nearby vessels;
(3) Collision avoidance: avoid collisions with nearby vessels.

3. Centralized formulation of the VTF problem

In this section, a centralized MPC method is proposed to solve the VTF problem. MPC has been popular in practical applications since its very early days (Mayne, 2014). At a particular time step, MPC solves an optimization problem to determine a sequence of actions that are expected to steer the system in the right direction; then, only the first action in the sequence is applied to the system; at the next time step, the optimization problem is solved again in a receding horizon fashion (Maestre and Negenborn, 2013; Negenborn and Maestre, 2014).

For waterborne transport, MPC also has many advantages. Firstly, the maneuverability of vessels is poor. The predictive property of MPC methods is beneficial to detect conflicts at an early stage. Secondly, as mentioned, the movement of a vessel is strongly influenced by external factors, such as wind, wave and current. Therefore, the real trajectories of the vessels are uncertain (Zheng et al., 2017). MPC considers the latest available measurement of the system's state and up-to-date information regarding disturbances, which is particularly helpful to deal with the uncertainties. MPC has been applied to vessel path following (Zheng et al., 2016), heading control (Li et al., 2012), and collision avoidance (Abdelaal et al., 2016; Zheng et al., 2017). Besides, distributed MPC has been used for cooperative control of networked vehicles (Dunbar and Murray, 2002; Richards and How, 2004; Keviczky et al., 2008; Kuwata and How, 2011; Muëller et al., 2012). Thirdly, research indicates that MPC has many advantages for the control of large-scale networked systems (Negenborn and Maestre, 2014). Therefore, in this paper, we consider MPC as a solution for the control of vessels in CMVSs.

For a single vessel, the task is smooth trajectory following with minimal energy consumption. Thus, the optimization problem solved by agent i to determine the actions of vessel i at each time step k is

$$\text{minimize} \quad J_i(\widetilde{u}_i(k)) = \sum_{\tau=1}^{H_\mathrm{p}} (\alpha\, \|y_i(k+\tau|k) - w_i(k+\tau)\|_p^2 + \gamma\, \|u_i(k+\tau-1|k)\|_p^2) \tag{10}$$

$$\text{subject to} \quad \text{the dynamics in (13) and constraints (5) and (6).} \tag{11}$$

where α and γ are the weights of trajectory following and control efforts; $w_i(k)$ is the reference trajectory; H_p is the prediction horizon; τ is the τth time step in the prediction horizon; $u_i(k+\tau\,|k)$ is the prediction of $u_i(k+\tau)$ at time k, the same goes for $x_i(k+\tau\,|k)$ and $y_i(k+\tau\,|k)$; $\widetilde{u}_i(k)$ indicates control input over the prediction horizon, i.e., $\widetilde{u}_i(k) = [u_i(k\,|k)'\cdots u_i(k+H_\mathrm{p}-1\,|k)']'$; similarly, the prediction of the states and outputs are expressed as

$$\widetilde{x}_i(k) = [x_i(k\,|k)'\cdots x_i(k+H_\mathrm{p}\,|k)']' = \widetilde{A}_i x_i(k) + \widetilde{B}_i \widetilde{u}_i(k), \tag{12}$$

$$\widetilde{y}_i(k) = [y_i(k\,|k)'\cdots y_i(k+H_\mathrm{p}\,|k)']' = \widetilde{C}_i x_i(k), \tag{13}$$

where $\widetilde{A}_i$, $\widetilde{B}_i$, and $\widetilde{C}_i$ are the prediction matrices.

In the optimization problem (10) and (11), the influences of the neighbors are not taken into account. However, a CMVS is a system in which vessels communicate and cooperate with others. The interactions between vessels are indispensable. Therefore, based on the three VTF rules, the objective function of each agent becomes

$$J_i(\widetilde{u}_i(k)) = \sum_{\tau=1}^{H_\mathrm{p}} \sum_{j\in N_i} (\alpha\, \|y_i(k+\tau\,|k) - w_i(k+\tau)\|_p + \beta\, \|d_{ij|i}(k+\tau\,|k) + \delta_{ij}(k+\tau\,|k)\|_p + \gamma\, \|u_i(k+\tau-1\,|k)\|_p) \tag{14}$$

where β is the weight of aggregation. $d_{ij|i}(k+\tau\,|k)$ is the distance between vessel i and vessels j calculated by vessel i with the position of j that i received ($p_{j|i}$), i.e.,

$$d_{ij|i}(k+\tau\,|k) = \|p_i(k+\tau\,|k) - p_{j|i}(k+\tau\,|k)\|_p. \tag{15}$$

To simplify the model, in this paper, at each time step k, the graph $G(k) = (\nu(k), e(k))$ is assumed to be constant over the prediction horizon. Then, the control problem for the cooperation in a CMVS is expressed as

$$\text{Problem } \mathscr{P}: \quad \text{minimize} \quad \sum_{i=1}^{n} J_i(\widetilde{u}_i(k)) \tag{16}$$

$$\text{subject to} \quad \forall\, i \in N, j \in N_i, \forall\, k \in T, \forall\, \tau \in H_{\mathrm{p}}:$$

$$u_{\min} \leqslant \|u_i(k+\tau|k)\|_2 \leqslant u_{\max}, \tag{17}$$

$$q_{\min} \leqslant \|q_i(k+\tau|k)\|_2 \leqslant q_{\max}, \tag{18}$$

$$d_{ij|i}(k+\tau|k) \geqslant d_{\mathrm{safe}}, \tag{19}$$

$$-r \leqslant \delta_{ij|i}(k+\tau|k) \leqslant r, \tag{20}$$

$$r = \min(r_1, r_2, \cdots, r_n). \tag{21}$$

According to Richards and How (2002), Problem $\mathscr{P}$ can be transferred into a mixed integer linear programming problem when $\|\cdot\|_p$ approaches the infinity norm $\|\cdot\|_\infty$. The method approximates velocity and input limitations with M linear constraints. Consequently, the collision avoidance, velocity and input constraints can be rewritten as

$$\forall\, k \in T, \forall\, \tau \in H_{\mathrm{p}}:$$

$$\begin{cases} d_{ij|i}^{x^+}(k+\tau\,|k) \geqslant d_{\mathrm{safe}} - \eta\phi_{ij|i,1}(k+\tau\,|k), \\ d_{ij|i}^{y^+}(k+\tau\,|k) \geqslant d_{\mathrm{safe}} - \eta\phi_{ij|i,2}(k+\tau\,|k), \\ d_{ij|i}^{x^-}(k+\tau\,|k) \geqslant d_{\mathrm{safe}} - \eta\phi_{ij|i,3}(k+\tau\,|k), \\ d_{ij|i}^{y^-}(k+\tau\,|k) \geqslant d_{\mathrm{safe}} - \eta\phi_{ij|i,4}(k+\tau\,|k), \\ \sum_{g=1}^{4} \phi_{ij|i,g}(k+\tau\,|k) \leqslant 3, \end{cases} \tag{22}$$

$$\begin{cases} \mathbf{1^M} q_{\min}\cos\left(\frac{\pi}{M}\right) \leqslant \Lambda q(k) \leqslant \mathbf{1^M} q_{\max}\cos\left(\frac{\pi}{M}\right), \\ \mathbf{1^M} u_{\min}\cos\left(\frac{\pi}{M}\right) \leqslant \Lambda u(k) \leqslant \mathbf{1^M} u_{\max}\cos\left(\frac{\pi}{M}\right), \end{cases} \tag{23}$$

where $d_{ij|i}^{x^+}(k+\tau\,|k)$, $d_{ij|i}^{y^+}(k+\tau\,|k)$, $d_{ij|i}^{x^-}(k+\tau\,|k)$, and $d_{ij|i}^{y^-}(k+\tau\,|k)$ are the distance between vessel i and j that vessel i measured in the four directions $+X$, $+Y$, $-X$, and $-Y$, respectively; η is a positive number that is much larger than any position or velocity to be encountered in the problem. $\phi_{ij|i,g}$ are the binary variables; $\mathbf{1^M}$ is an $M \times 1$ column vector with all components equal to 1. Λ is the approximation matrix:

$$\Lambda = \begin{bmatrix} \sin\left(\frac{2\pi m}{M}\right) & \cos\left(\frac{2\pi m}{M}\right) \\ \vdots & \vdots \\ \sin(2\pi) & \cos(2\pi) \end{bmatrix}, \quad m \in \mathbb{Z}, 1 \leqslant m < M. \tag{24}$$

4. Distributed formulation of the VTF problem

As mentioned in several studies (Maestre and Negenborn, 2013; Negenborn and Maestre, 2014; Zheng et al., 2016), the centralized controller does not scale well. The increasing number of decision variables, state variables, and measurements can significantly increase the computational time required to determine actions (Negenborn and Maestre, 2014). Below, Problem $\mathscr{P}$ is decomposed into subproblems that are solved by the vessels locally in combination with information exchange. A serial iterative DMPC approach based on the ADMM is proposed.

In the distributed framework, each vessel makes decisions based on the information that other vessels broadcast. When the vessels have reached an agreement, the solution that each vessel calculates in iteration s should be the same as the trajectory it broadcast in iteration $s-1$. Therefore, we introduce a common global variable z to record the trajectories that vessels broadcast, $z(k) = [z_1(k)' \cdots z_n(k)']'$. Since all the vessels are considered as linear time-invariant systems, $z_i(k)$ is a copy of the input that vessel i determined for previous iteration, i.e., $z_i(k)^s = \widetilde{u}_i(k)^{s-1}$. Therefore, Problem $\mathscr{P}$ can be rewritten with local variable u_i and global variable z:

$$\text{minimize} \quad \sum_{i=1}^{n} J_i(\widetilde{u}_i(k)) \tag{25}$$

$$\text{subject to} \quad \widetilde{u}_i(k) = z_i(k), \forall\ i \in N \text{ and constraints (17)-(21).} \tag{26}$$

The information that the vessels broadcast are their predictive trajectories over the prediction horizon. In this paper, we assume that the bandwidth is sufficient, and no delay in communication among vessels in a CMVS.

4.1. ADMM decomposition

Among the decomposition techniques, ADMM is one of the widely applied methods to solve the distributed MPC problems (Zheng et al., 2016; Farokhi et al., 2014). ADMM is an algorithm that is intended to blend the decomposability of dual ascent with the superior convergence properties of the method of multipliers (Boyd et al., 2011). The first step of this method is to form the augmented Lagrangian of the optimization problem; then, the primal and the dual variable are updated in an alternating or sequential fashion; the iteration stops when a stopping criterion is met.

Therefore, we firstly construct the augmented Lagrangian for the problem (25) and (26). In this paper, we employ quadratic penalty terms, because currently no proof of convergence is known for ADMM with non-quadratic penalty terms (Boyd et al., 2011):

$$L([(\widetilde{u}_i(k)]_{i=1}^{n}, [z_i(k)]_{i=1}^{n}, [\lambda_i(k)]_{i=1}^{n}, [\rho_i]_{i=1}^{n}) = \sum_{i=1}^{n} (J_i(\widetilde{u}_i(k)) + \lambda_i(k)'(\widetilde{u}_i(k) - z_i(k)) + \rho_i/2\ \|\widetilde{u}_i(k) - z_i(k)\|_2^2), \tag{27}$$

where ρ_i is penalty parameter and $\lambda_i(k)$ is a dual variable. Then, ADMM consists of the iterations to update variables:

$$\widetilde{u}_i^s(k) = \underset{\widetilde{u}_i(k)}{\operatorname{argmin}} (J_i(\widetilde{u}_i(k)) + \lambda_i^{s-1}(k)'(\widetilde{u}_i(k) - z_i^{s-1}(k)) + \rho_i/2\ \|\widetilde{u}_i(k) - z_i^{s-1}(k)\|_2^2), \tag{28}$$

$$z_i^s(k) = \widetilde{u}_i^s(k) + \lambda_i^{s-1}(k)/\rho_i, \tag{29}$$

$$\lambda_i^s(k) = \lambda_i^{s-1}(k) + \rho_i(\widetilde{u}_i^s(k) - z^s(k)). \tag{30}$$

where $\cdot^s$ is the value of corresponding variable at the sth iteration.

Different from the conventional form of ADMM, for vessel i, the update of the corresponding part in global variable z_i only needs information of itself. Therefore, here, *all the updating of variables can be carried out independently for each vessel i*. The iterations stop when the primal ($R_{\text{pri},i}^s$) and dual residuals ($R_{\text{dual},i}^s$) meet the criteria:

$$\|R_{\text{pri},i}^s\|_2 = \|\widetilde{u}_i^s(k) - z^s(k)\|_2 \leqslant \varepsilon_{\text{pri},i}^s, \tag{31}$$

$$\|R_{\text{dual},i}^s\|_2 = \|z_i^s(k) - z_i^{s-1}(k)\|_2 \leqslant \varepsilon_{\text{dual},i}^s, \tag{32}$$

where $\varepsilon_{\text{pri},i}^s$ and $\varepsilon_{\text{dual},i}^s$ are the tolerance. They usually can be determined by following equation:

$$\begin{aligned} \varepsilon_{\text{pri},i}^s &= \sqrt{Nn_u}\,\varepsilon^{\text{abs}} + \varepsilon^{\text{rel}}\max\{\|\widetilde{u}_i^s(k)\|_2,\ \|z_i^s(k)\|_2\}, \\ \varepsilon_{\text{pri},i}^s &= \sqrt{Nn_u}\,\varepsilon^{\text{abs}} + \varepsilon^{\text{rel}}\ \|\lambda_i(k)^s\|_2, \end{aligned} \tag{33}$$

where ε^{abs} and ε^{rel} are the absolute tolerance and relative tolerance, respectively.

The tolerance means that when the iteration stops, there is still a difference between $u_i(k)$ and $z_i(k)$. Although the difference is small, it may make the trajectories that a vessel actually choose deviate from the trajectories it sends to others. To guarantee that the collision avoidance constraints are satisfied, we make an adjustment on the safety distance over the prediction horizon. We find out the largest trajectory deviation under the worst situation in which the deviation of the first control input equals to the tolerance, i.e.,

$$u_{d_{\text{fix}}} = \left[d_{\text{fix}} \quad d_{\text{fix}} \quad \underbrace{0 \ \cdots \ 0}_{2\times(H_{\text{p}}-1)} \right]', \tag{34}$$

$$d_{\text{fix}} = \sqrt{Nn_u}\,\varepsilon^{\text{abs}} + \varepsilon^{\text{rel}}\ \|u_{\text{max}}\|. \tag{35}$$

Then, the deviation is added to the safety distance:

$$\widetilde{d}_{\text{safe}} = \mathbf{1}^{H_{\text{p}}} d_{\text{safe}} + \widetilde{B}_i u_{d_{\text{fix}}}. \tag{36}$$

Therefore, even if it is the worst situation, the collision avoidance constraints still can be met.

As suggested by Boyd et al. (2011), He et al. (2000) and Wang and Liao (2001), a varying penalty parameter helps to improve the convergence in practice, as well as making performance less dependent on the initial choice of the penalty parameter. Therefore, in this paper, the scheme proposed in He et al. (2000) is adopted:

$$\rho_i^s = \begin{cases} 2\rho_i^{s-1} & \text{if } \|R_{\text{pri},i}^s\|_2 > 10\ \|R_{\text{dual},i}^s\|_2, \\ \rho_i^{s-1}/2 & \text{if } \|R_{\text{dual},i}^s\|_2 > 10\ \|R_{\text{pri},i}^s\|_2, \\ \rho_i^{s-1} & \text{otherwise.} \end{cases} \tag{37}$$

4.2. Serial iterative ADMM-based DMPC

In the literature, two frameworks for cooperative DMPC are employed, non-iterative and iterative (Scattolini, 2009). With the non-iterative framework, the subsystems optimize their performance criteria in a sequential order. Since information is exchanged only once after an agent solved its problem, the amount of communication between agents is less, as well as the computation time. In Kuwata and How (2011), the authors solve the trajectory optimization problem with the non-iterative framework. Vehicles solve their own problems in sequence while generating feasible modifications to the prediction of other vehicles' plan. The fleet objective can be improved by having some vehicles sacrifice their individual objectives.

Alternatively, in the iterative framework, the agents obtain agreement through a number of iterations. By exchanging the information about its own decision and other agents' preferences, the inputs will converge, and a set of actions for all agents will be found. Thus, iterative frameworks have a larger potential to achieve overall optimal performance (Negenborn and Maestre, 2014). According to the communication schemes, the iterative DMPC is divided into two types: parallel and serial (Negenborn and Maestre, 2014; Scattolini, 2009).

The parallel schemes are frequently used throughout the literature. In parallel schemes, all the agents perform computations at the same time. Because of the potential conflicts of objectives, the solutions may be not converged. To solve the problem, some research considers a two-level hierarchical control structure: an algorithm at the higher level coordinates the action of local controllers placed at a lower level, such as (Zheng et al., 2016). A parallel iterative scheme with a single-layer control structure has also been presented in the past. In the algorithm with single-layer control structure proposed by Farokhi et al. (2014), each subsystem computes optimal inputs for itself and all its neighbors. At each time step, the actions are determined by solutions calculated by the vessel itself and its neighbors.

On the contrary, in the serial schemes, only one agent is performing computations at a time. Serial schemes have the advantage over the parallel schemes that agents make use of the most up-to-date information from their neighbors. In Negenborn and Maestre (2014), a serial iterative scheme for transport networks is proposed. It shows that the serial scheme has preferable properties in terms of solution speed, by requiring fewer iterations, and solution quality.

In the following part, we propose a serial iterative algorithm for the VTF problem of CMVSs. Compared with existing algorithms, the control actions are computed in a truly distributed way. Each vessel only computes the input for itself, according to its own state and the information its neighbors broadcast.

In the vessel train formation problem, the information that a vessel needs to make decisions is the predictive trajectories of its neighbors. Thus, with the serial iterative scheme, $p_{ij|i}$ in (15) and (22) updates as follows:

$$p_{ij|i}^s(k+\tau|k) = \begin{cases} p_j^s(k+\tau|k) & j \in N_i, j \text{ solves its problem before } i, \\ p_j^{s-1}(k+\tau|k) & j \in N_i, j \text{ solves its problem after } i. \end{cases} \tag{38}$$

To summarize, at each time step k, iteration s, the proposed algorithm is carried out by following steps:

Step 1: Vessel i determines $\tilde{u}_i^s(k)$ by solving its local subproblem;
Step 2: Vessel i updates the corresponding global variable, Lagrange multiplier and send the information to the next Vessel j;
Step 3: Vessels repeat Step 1 and 2 until all the vessels finish computation;
Step 4: Vessels move to the next iteration $s+1$, and repeat Step 1–3. The iterations stop when the stopping condition is satisfied;
Step 5: Each vessel implements the actions until the beginning of the next control cycle.

A detailed description of the algorithm is shown in Algorithm 1. This algorithm may fail if an obstacle is within an inevitable collision distance when it is detected at the first time. In fact, this kind of obstacles is also difficult or even impossible to avoid in current human-operated practice. Therefore, we make following assumptions: (1) the initial state of a vessel is feasible; (2) when an obstacle is detected by a vessel at the first time, it is avoidable (for details about inevitable collision states, the interested reader can refer to Parthasarathi and Fraichard (2007), Fraichard and Asama (2003)).

Algorithm 1. Serial iterative ADMM-based DMPC

Algorithm 1: Serial iterative ADMM-based DMPC

while $k \leqslant T$ **do**
    $z^0(k) := \widetilde{u}_i^0(k) := \left[\widetilde{u}_i^{\text{end}}(k : \text{end}\,|k-1); 0\right]; \quad \lambda_i^0(k) := 0; \quad \rho := \rho_{\text{ini}};$
    for $s = 1 : S$ **do**
        $jdg := 0; \quad N_{\text{jump}} := 0;$
        for $i = 1 : N$ **do**
            // Each subcontroller solves the local subproblem
            $$\widetilde{u}_i^s(k) := \underset{\widetilde{u}_i(k)}{\arg\min} \Big(J_i\left(\widetilde{u}_i(k)\right) + \lambda_i^{s-1}(k)' \left(\widetilde{u}_i(k) - z_i^{s-1}(k)\right) + \rho_i/2 - \left\|\widetilde{u}_i(k) - z_i^{s-1}(k)\right\|_2^2 \Big);$$
            if *solution does not exit* † **then**
                $\widetilde{u}_i^s(k) := \widetilde{u}_i^{s-1}(k); \quad N_{\text{jump}} := N_{\text{jump}} + 1;$
            // Update global variable and Lagrange multipilier
            $z_i^s(k) := \widetilde{u}_i^s(k) + \lambda_i^{s-1}(k) \Big/ \rho_i$;
            $\lambda_i^s(k) := \lambda_i^{s-1}(k) + \rho_i\left(\widetilde{u}_i^s(k) - z^s(k)\right);$
            $ZX_i^s := \tilde{A}_i x_i(k) + \tilde{B}_i z_i(k);$
            // Update primal and dual residual and tolerance
            $R_{\text{pri},i}^s := \widetilde{u}_i^s(k) - z^s(k); \quad R_{\text{dual},i}^s := z_i^s(k) - z_i^{s-1}(k);$
            $\varepsilon_{\text{pri},i}^s := \sqrt{N n_u}\varepsilon^{\text{abs}} + \varepsilon^{\text{rel}} \max\left\{\|\widetilde{u}_i^s(k)\|_2, \|z_i^s(k)\|_2\right\};$
            $\varepsilon_{\text{dual},i}^s := \sqrt{N n_u}\varepsilon^{\text{abs}} + \varepsilon^{\text{rel}}\|\lambda_i(k)^s\|_2;$
            // Stopping check
            if $\left\|R_{\text{pri},i}^s\right\|_2 \leqslant \varepsilon_{pri,i}^s$ *and* $\left\|R_{\text{dual},i}^s\right\|_2 \leqslant \varepsilon_{\text{dual},i}^s$ **then**
                $jdg := jdg + 1;$
            // Update the penalty parameter
            case $\left\|R_{\text{pri},i}^s\right\|_2 > 10\left\|R_{\text{dual},i}^s\right\|_2$ **do** $\rho_i := 2\rho_i;$
            case $\left\|R_{\text{dual},i}^s\right\|_2 > 10\left\|R_{\text{pri},i}^s\right\|_2$ **do** $\rho_i := \rho_i/2;$
            // Send ZX_i^s, jdg and N_{jump} to other agents
        if $jdg = N$ *and* $N_{\text{jump}} = 0$ **then** break;
    // Update the state of each vessel and move to next step

† When a vessel fails to find a solution at iteration s, it will keep its solution at iteration $s-1$. Other vessels will try to find the collision-free solutions. If the vessel still cannot find solutions at iteration $s + 1$ (i.e., other vessels cannot find solutions, either), one of the following actions will be taken: (1) a centralized controller will be activated, and calculates the trajectories for all vessels; (2) all the vessel will slow down to postpone conflicts until they find solution. Thanks to the assumptions on initial states of the vessels and the obstacles, this situation never occurred in the simulation experiments shown in this paper.

5. Simulation experiments

In the serial iterative approach, a vessel assumes that the vessels updating later keep their previous trajectories. Thus, it should take actions to meet all the constraints. Therefore, the serial iterative approach can be regarded as giving priorities to the vessels updating later. Thus, the updating sequence influences the final solutions. Moreover, vessels make decisions based on the information that others send. One vessel can broadcast a trajectory closer to its original one to share the responsibility for collision avoidance.

In this section, simulation experiments are carried out to analyze the impacts of different information updating sequences and so-called responsibility parameters. Moreover, we analyze scalability of the proposed method. After that, a simulation of a CMVS navigating from different terminals in the Port of Rotterdam to inland waterways is carried out to show the potential of the proposed algorithm.

The experiments are carried out with Matlab 2016a. The optimization problems of the controllers are solved by ILOG CPLEX

Table 1
Simulation set ups.

u_{max}	u_{min}	p_{max}	p_{min}	M	d_{safe}	r
1 m/s^2	0 m/s^2	3 m/s	0 m/s	16	10 m	20 m

H_p	α	β	γ	ε^{abs}	ε^{rel}
10	10	0	1	10^{-3}	10^{-3}

Optimization Studio (Version 12.6.3). The experiments are run on a PC with a dual-core 3.2 GHz Intel(R) Core(TM) i5-3470U CPU and 8 GB of RAM.

5.1. Influencing factors in the serial iterative algorithm

In Sections 5.1.1 and 5.1.2, a head-on scenario of two vessels is simulated to analyze the impacts of information updating sequences and responsibility parameters. In the simulation, Vessel 1 is from (0, 0) to (0, 300), and Vessel 2 is from (0, 300) to (0, 0). Section 5.1.3 provides the relations between computational time, number of iterations and the number of vessels in a CMVS. The setup of the simulation experiments is listed in Table 1.

5.1.1. Impacts of updating sequence

We test 4 updating sequences: (a) in order ($1 \rightarrow 2$); (b) in reverse ($2 \rightarrow 1$); (c) alternately ($1 \rightarrow 2 \rightarrow 2 \rightarrow 1$); (d) randomly choose from ($1 \rightarrow 2$) and ($2 \rightarrow 1$). The results are analyzed below.

Fig. 3 zooms in the trajectories of the two vessels under the head-on situation with different updating sequences. The results are compared with the trajectories when using a centralized controller. When the vessels update in order or in reverse, one vessel gives ways to the other. With the alternative order and the random order, both vessels have to take actions. However, the performance of the algorithm with a random order is uncertain. Sometimes it is the one closest to the centralized controller, while sometimes it can be the one with the worst performance. In Fig. 3, the trajectories of vessels in the case with the random order is the results of one experiment. The results of the case exchanging randomly in Figs. 4–7 are the average of results of 20 experiments.

The local and overall costs are shown in Fig. 4. In the cases updating in order and updating in reverse, the overall costs are almost the same while the local costs are swapped. The local costs of the vessels in the case when they iterate alternately is closer to each other than the above two cases. However, the vessel that computing earlier at the first time step still has higher costs. Moreover, the total costs increase in the case updating iteratively.

Figs. 5 and 6 show that under all the situations, the inputs and collision avoidance constraints are all met. Vessels frequently take actions during 60–90 s when the collision avoidance occurs. Fig. 7 provides the information about computational time and the number of iterations needed in each time steps during 60–90 s. With the algorithms which update iteratively and randomly, more iterations are needed to reach an agreement. As a consequence, longer computational time is needed.

To sum up, with the proposed serial iterative algorithm, the vessel which computes earlier should give priority to the vessels which update later. Applying the alternative or random order can reduce the inequality, but more iterations are needed as a sacrifice.

5.1.2. Impacts of responsibility

It is worth to note that the trajectories that a vessel broadcasts is not necessary to be the solution of the optimization problem. Vessels can share the responsibility of collision avoidance by broadcasting trajectories that they prefer. Therefore, we introduce a

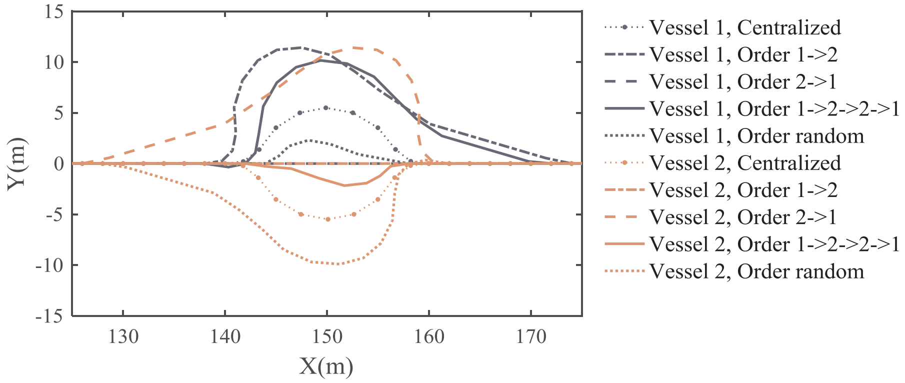

Fig. 3. Trajectories of each vessel with different updating sequences.

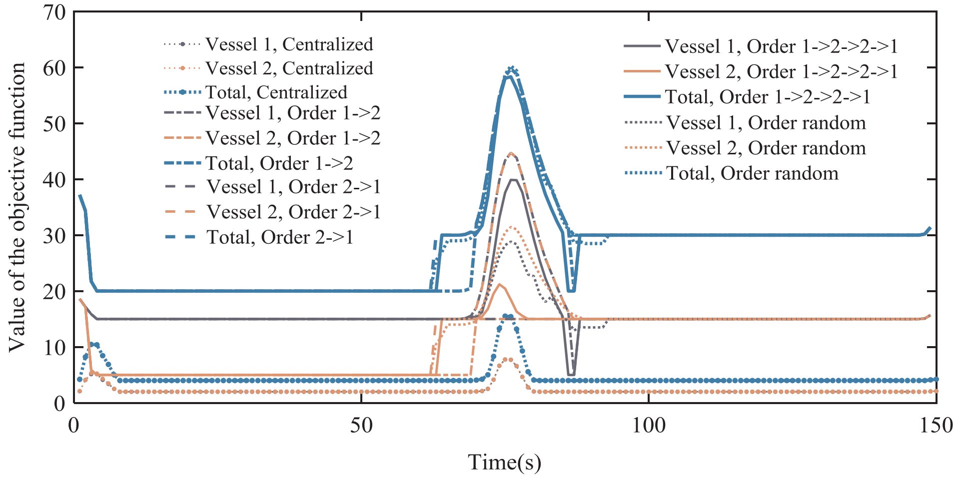

Fig. 4. Objective value of each vessel with different updating sequences.

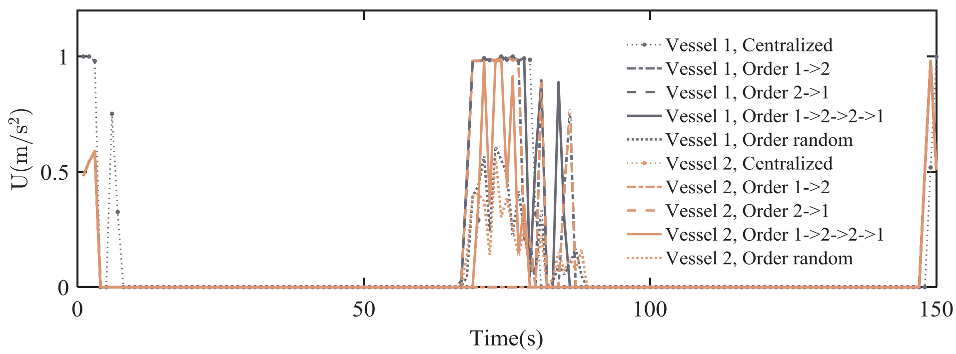

Fig. 5. Control input of each vessel with different updating sequences.

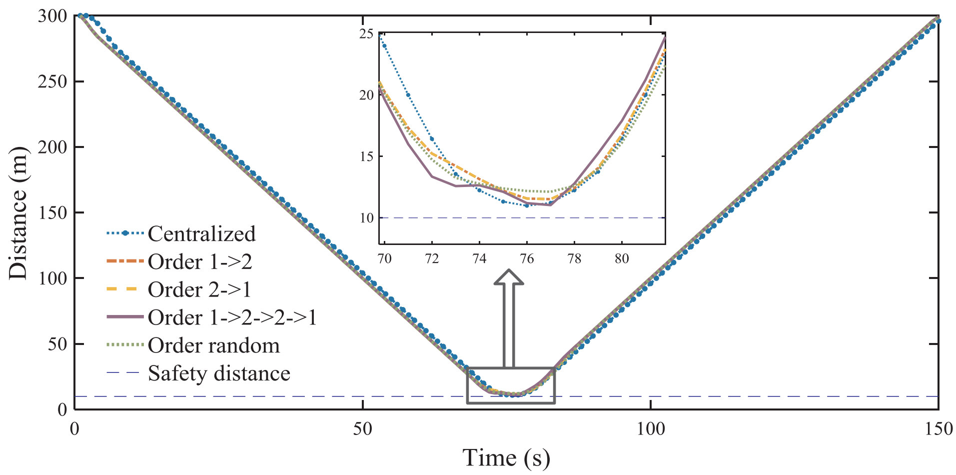

Fig. 6. Relative distance with different updating sequences.

responsibility parameter φ_i in the global variable updating (Line 11 in Algorithm 1). Thus, the global variable z_i is updated as follows:

$$z_i^s(k) := \varphi_i u_i^s(k) + (1-\varphi_i) z_i^{s-1}(k) + \lambda_i^{s-1}(k)/\rho_i, \quad \sum_{i=1}^{n} \varphi_i \geqslant 1, \quad 0 \leqslant \varphi_i \leqslant 1. \tag{39}$$

A smaller responsibility parameter φ_i indicates that the vessel is more willing to keep its original trajectory. Therefore, at each iteration, the predictive trajectory it sends to others is closer to its original one. Fig. 8 shows the broadcast trajectories of two vessels

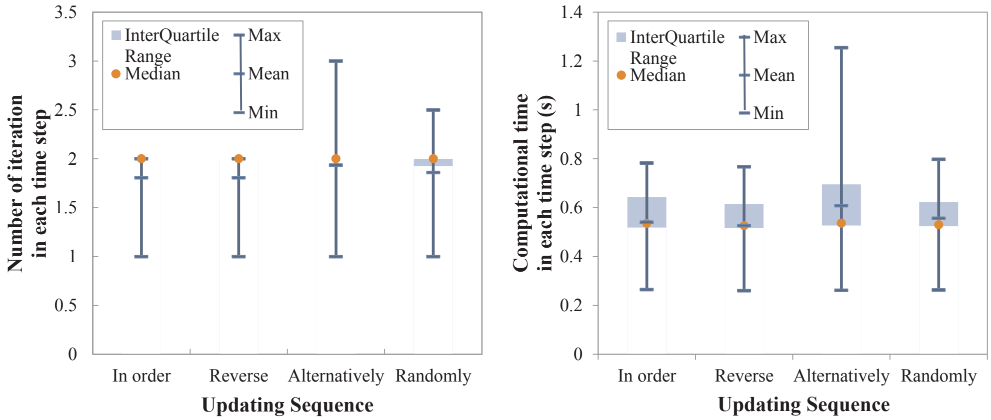

Fig. 7. Number of iterations and computational time in each time step with different updating sequences.

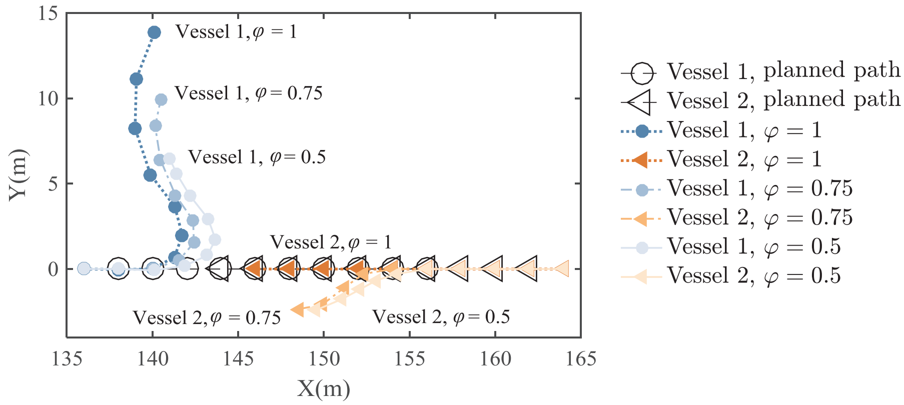

Fig. 8. Broadcast trajectories of the vessels using different responsibility parameters (1st iteration).

when they encountered. With $\varphi_1 = \varphi_2 = 1$, the earlier updated vessel (Vessel 1) has to take actions for collision avoidance, while Vessel 2 can keep its planned path. When φ_i decreases, the trajectory that Vessel 1 broadcasts does not satisfy the safety constraint. Therefore, Vessel 2 has to deviate from its original path. The trajectory that Vessel 2 provides might still not avoid the collision, so next iteration starts. Fig. 9 shows how safety at $k = 67$ is achieved through iterations. The smaller φ_i is, the more iterations are needed to reach an agreement.

The simulation results of the experiments with different responsibility parameters are shown in Figs. 10 and 11. In the simulation, $\varphi_1 = \varphi_2 = \varphi$. With a smaller responsibility parameter, the number of iterations needed in each time step increases significantly. In

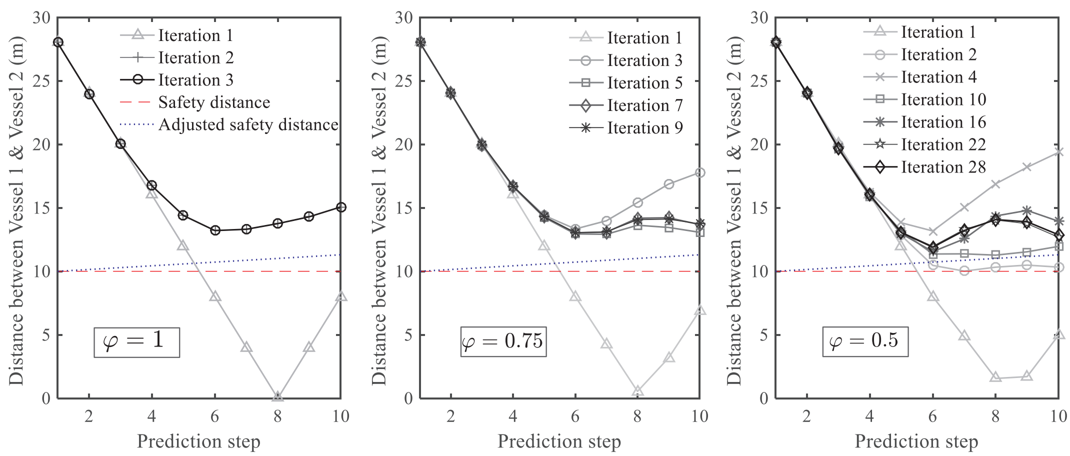

Fig. 9. Predicted distance evolution.

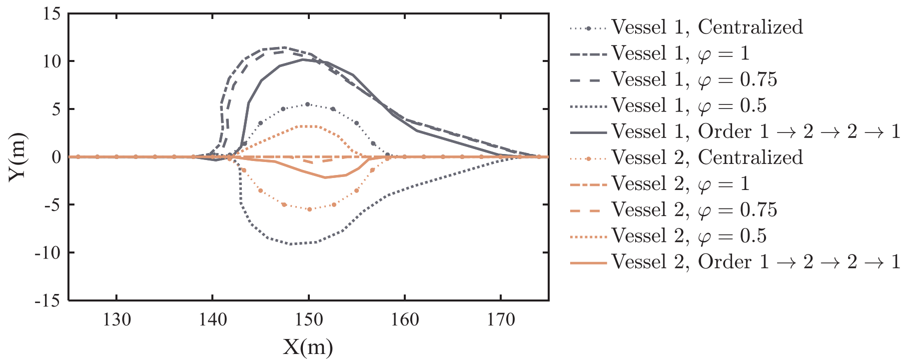

Fig. 10. Trajectories of vessels using different responsibility parameters.

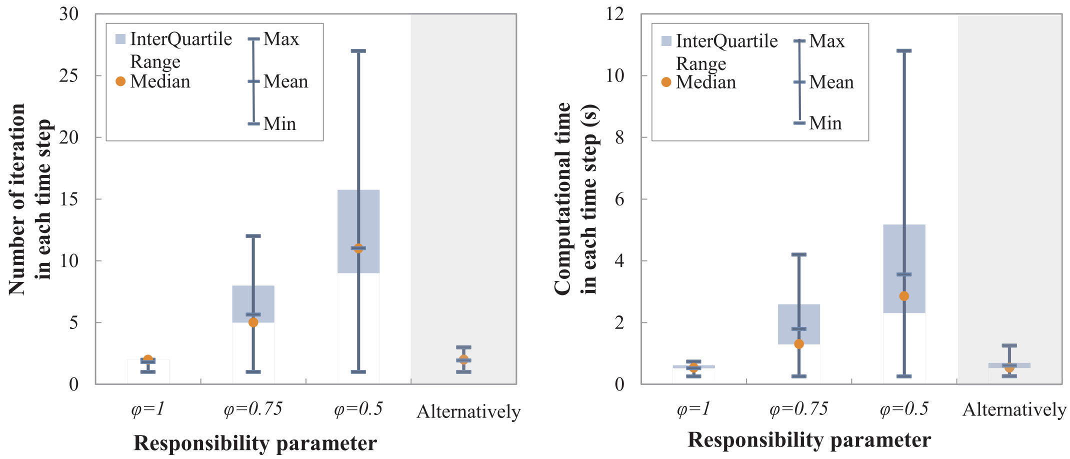

Fig. 11. Number of iterations and computational time with different responsibility parameters.

Fig. 11, the number of iterations and computational time of the algorithm updating iteratively are also provided. Compared with the case with alternative updating order, the computational time is much longer in the case with a small responsibility parameter. Therefore, we chose the alternatively updating algorithm for the subsequent experiments.

5.1.3. *Scalability*

To have an insight into the scalability of the proposed algorithm, we carried out several simulation experiments. In the simulation,

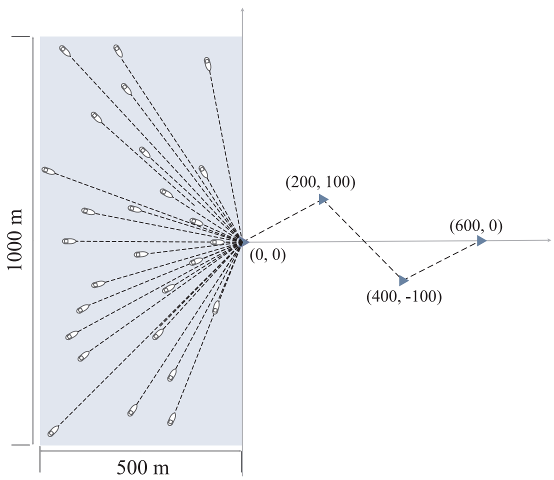

Fig. 12. Reference paths in the scalability tests.

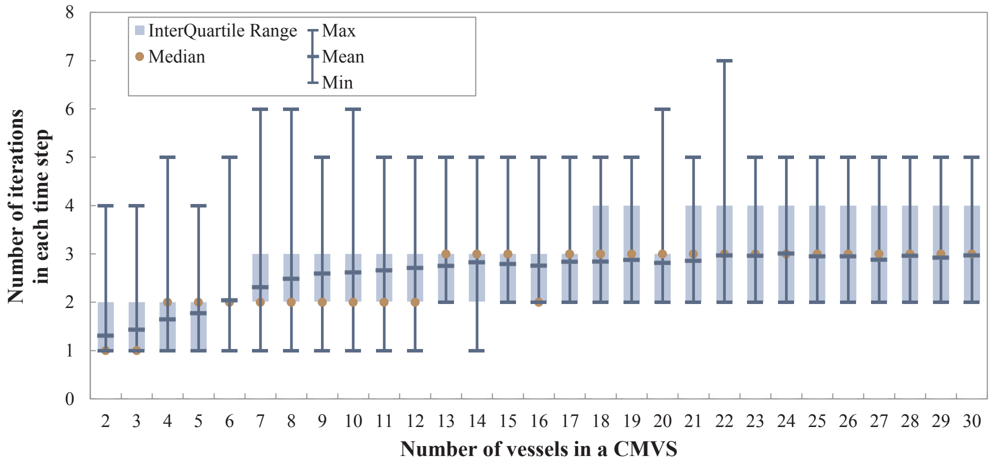

Fig. 13. Number of iterations when the number of vessels in a CMVS increases.

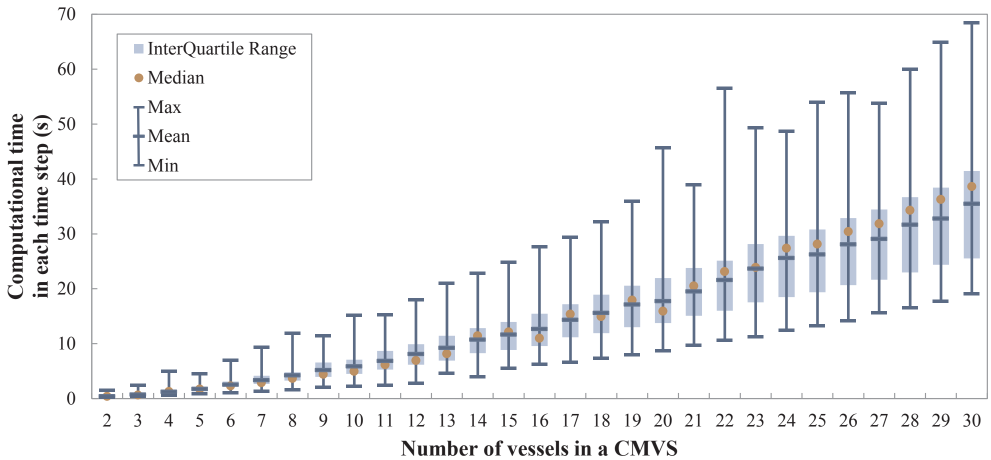

Fig. 14. Computational time when the number of vessels in a CMVS increases.

vessels follow the paths shown in Fig. 12. The number of vessels in a CMVS increases from 2 to 30. A vessel joints the CMVS once it arrives at (0, 0).

The minimum, maximum, mean value and median of the number of iterations and computational time at each time step are presented in Figs. 13 and 14. Both the number of iterations and computational time show growing tendencies. However, the increase in iterations is gradual. Iterations are used to solve the conflicts among the vessels. Because of the train-like formation, the conflicts do not increase even though the number of vessel increases. On the contrary, because the number of constraints in the Problem $\mathscr{P}$ increase with the number of formation mates, the computational time increases significantly. However, even when the number of vessels in a CMVS is up to 30, the maximum computation time is less than 70 s. Moreover, as we mentioned, the vessels in a CMVS are (temporally) have the same path or can pass through locks and bridges together. Therefore, the number of vessels in a CMVS is usually less than 10.

5.2. Simulation of a CMVS

In this part, a simulation of a CMVS consists of 5 vessels navigating from the different terminals in the Port of Rotterdam to inland waterways is presented.

5.2.1. Simulation setup

The simulation area is shown in Fig. 15. Five vessels start from different terminals (O_1–O_5), and they navigate together through the inland waterways. The vessels have reference paths which are indicated by waypoints. The position of the origins, waypoints and the destination are listed in Table 2. CW means conman waypoints. In the simulation, vessels arrive at the first conman waypoint CW_1 with a certain time interval (10s).

The algorithm that we use is the serial iterative ADMM-based DMPC presented in Section 4. The update order is iteratively from the first to the last and from the last to the first. Parameters needed in the simulation are given in Table 3.

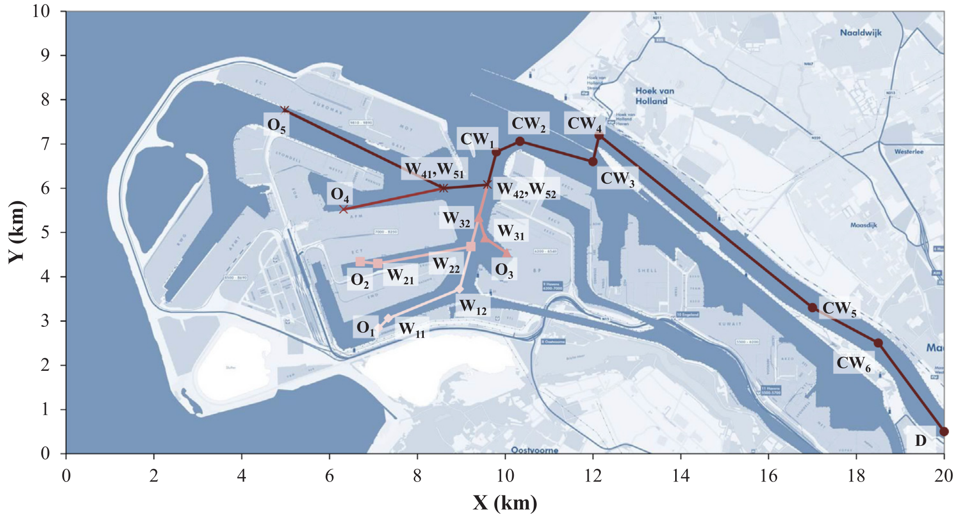

Fig. 15. Simulation area Port of Rotterdam Authority (2017).

Table 2
Origins, destination and waypoints in the simulation.

Node	X	Y	Node	X	Y	Node	X	Y	Node	X	Y
$\mathbf{O_1}$	7.12	2.84	$\mathbf{W_{11}}$	7.34	3.06	$\mathbf{W_{12}}$	8.96	3.72	$\mathbf{CW_3}$	12	6.6
$\mathbf{O_2}$	6.7	4.34	$\mathbf{W_{21}}$	7.1	4.3	$\mathbf{W_{22}}$	9.22	4.68	$\mathbf{CW_4}$	12.14	7.2
$\mathbf{O_3}$	10.04	4.54	$\mathbf{W_{31}}$	9.54	4.88	$\mathbf{W_{32}}$	9.4	5.34	$\mathbf{CW_5}$	17	3.3
$\mathbf{O_4}$	6.32	5.52	$\mathbf{W_{41}}$, $\mathbf{W_{51}}$	8.6	6	$\mathbf{CW_1}$	9.8	6.82	$\mathbf{CW_6}$	18.5	2.5
$\mathbf{O_5}$	4.98	7.76	$\mathbf{W_{42}}$, $\mathbf{W_{52}}$	9.6	6.08	$\mathbf{CW_2}$	10.34	7.06	$\mathbf{D}$	20	0.5

Table 3
Simulation setup.

u_{max}	u_{min}	p_{max}	p_{min}	M	d_{safe}	r
2 m/s^2	0 m/s^2	6 m/s	0 m/s	16	40 m	250 m

H_p	α	β	γ	ε^{abs}	ε^{rel}	Sampling
8	10	1	1	10^{-3}	10^{-3}	10 s

5.3. Simulation results

The trajectories of the five vessels are shown in Fig. 16. All the vessels follow the reference paths. Vessels deviate from the given path at the bend waterways to avoid collisions.

Fig. 17 shows the simulation results. At the bends, vessels need to take actions frequently, such as $k = 150$ and $k = 280$. Due to the effective communication, vessels can timely respond to the velocity changes that others make. As a consequence, although the distances between vessels are fluctuating when vessels navigate through the bends, they are always larger than safety distance. After that, when vessels enter the straight segment, the speed of the vessels become consistent, and the distances between the vessels become constant.

The number of iterations and computational time at each time step are provided in Fig. 17. At each time step, vessels can find the solution within 10 iterations. The computational time keeps the same pattern with the number of iterations: when the number of iterations increases, the computational time increases. The computational time is less than 10s, which means that the optimization problem is solved within the sampling time. What worth to mention is that, when the vessel train is formed, fewer input changes, fewer iterations and therefore, less computational time are needed. For example, from $k = 160$ to $k = 240$, the control input is 0 and only 1 iteration is needed.

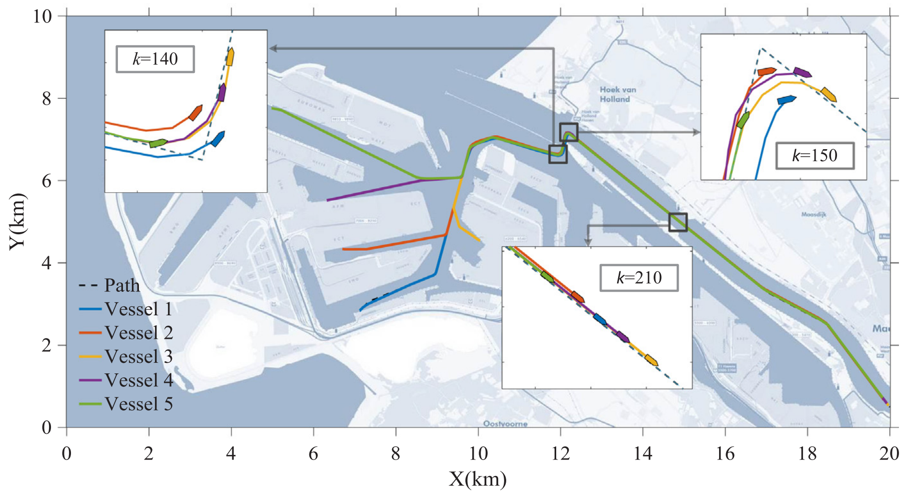

Fig. 16. Trajectories of the vessels.

6. Conclusions and future research

6.1. Conclusions

This paper focuses on the cooperative control of multiple vessels. We propose the concept of a Cooperative Multi-Vessel System (CMVS) consisting of coordinated autonomous vessels. Vessels in a CMVS have to follow predetermined paths and avoid collisions with others. To gain the benefits of sailing in groups, vessels attempt to stay close to nearby vessels. The cooperation of vessels in a CMVS is formulated as a Vessel Train Formation (VTF) problem. In this article, an MPC-based method for addressing the VTF problem is proposed. The control problem is decomposed into a distributed form using ADMM. A single-layer serial iterative structure is applied in the distributed formulation: each vessel employs MPC to determine its actions; the agreements among the vessels are achieved via serial iterative negotiations.

Simulations are carried out to analyze the effects of varing information updating sequences on overall and individual solutions. Due to the nature of the serial iterative scheme, the vessels that update earlier have less up-to-date information and thus make less optimal decisions. Applying the alternative or random order can reduce the inequality, although at the price of more iterations and computational time. Moreover, we introduce a responsibility parameter to indicate the willingness of trajectory change. The larger the parameter is, the more responsibility of collision avoidance a vessel bears, the bigger changes it accepts to make to its trajectories. Similarly, a smaller responsibility parameter helps to balance the changes that the vessels make, while the number of iterations and computational time increase considerably. The scalability of the proposed algorithm is analyzed as well. The increase of the number of vessels has more impacts on computational time than the number of iterations. The train-liked formation reduces the demand on collision avoidance. Consequently, the proposed method is capable to solve the VTF problem online within the sampling time when the number of vessels in a CMVS increases to 30. To show the potential of our method, we further simulate the scenario in which a CMVS consisting of five vessels navigates from the Port of Rotterdam to inland waterways. The proposed method successfully steers the vessels from different origins to form a vessel train. Due to the effective communication, vessels can timely respond to the velocity changes that others make. After the train-like formation is formed, the speed of the vessels become consistent, and the distances between vessels become constant. Thus, CMVSs have the potential to enhance the safety of waterborne transport systems.

6.2. Future research directions

Although substantial efforts have been made to achieve successful cooperation strategies, cooperative control of vessels is a challenging topic. Future research should focus on the following aspects:

- Theoretical analysis
  The proof of the recursive feasibility and stability is not considered in this paper, as we focus more on the application. As mentioned in Mayne (2014), stability and recursive feasibility of MPC is usually achieved in two different ways: imposing conditions on the terminal cost and/or constraint set, and extending the horizon. However, a formal recursive feasibility and stability analysis of distributed MPC is still challenging.
- Environmental disturbances and uncertainties
  As mentioned in earlier literature (Ihle, 2006; Almeida et al., 2010; Zheng et al., 2017), the dynamics of vessels are strongly influenced by environmental disturbances, e.g., wind, waves, and currents. Therefore, there will be uncertainties in the predicted trajectories. Future research will address how to deal with those uncertainties, e.g., by considering the integration of Robust MPC ideas (Zheng et al., 2017).

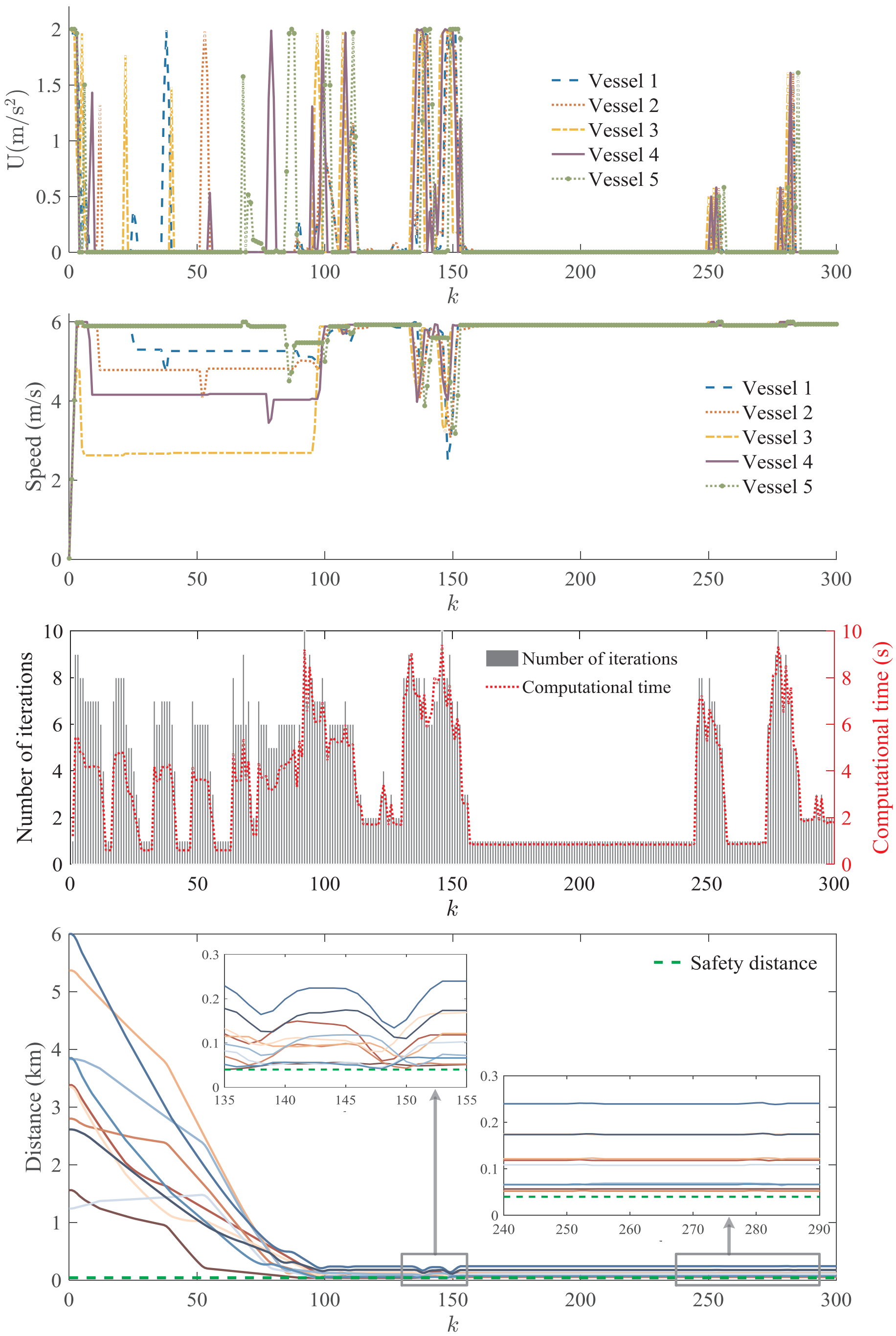

Fig. 17. Simulation results.

- Intra-CMVS interaction and Vessel-to-Infrastructure cooperation
  This paper concentrates on the cooperation of vessels in a CMVS. Cooperation between CMVSs, in particular the encountering of CMVSs at the intersections, will be studied in future research. Besides, locks and bridges are usually bottlenecks (Economic Commission for Europe, 2013) in inland waterway networks. Therefore, Vessel-to-Infrastructure cooperation is also an important aspect of the research on CMVSs.
- Physical experiments
  Many control strategies have been proposed for individual autonomous vessels. However, most of them are only validated by simulations or in a laboratory environment. In order to apply CMVSs in reality, it is important to assess the potential with physical experiments.

Acknowledgment

This research is supported by the China Scholarship Council under Grant 201406950041.

References

Abdelaal, M., Fränzle, M., Hahn, A., 2016. Nmpc-based trajectory tracking and collison avoidance of underactuated vessels with elliptical ship domain. IFAC-PapersOnLine 49 (23), 22–27.
Almeida, J., Silvestre, C., Pascoal, A., 2010. Cooperative control of multiple surface vessels in the presence of ocean currents and parametric model uncertainty. Int. J. Robust Nonlinear Control 20 (14), 1549–1565.
Arrichiello, F. Chiaverini, S., Fossen, T.I., 2006. Formation control of underactuated surface vessels using the null-space-based behavioral control. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5942–5947.
Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. 3 (1), 1–122.
Campbell, S., Naeem, W., Irwin, G., 2012. A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres. Annu. Rev. Control 36 (2), 267–283.
Daily, R., Bevly, D.M., 2008. Harmonic potential field path planning for high speed vehicles. In: Proceedings of 2008 American Control Conference, pp. 4609–4614.
Dunbar, W.B., Murray, R.M., 2002. Model predictive control of coordinated multi-vehicle formations. In: Proceedings of the 41st IEEE Conference on Decision and Control, vol. 4. IEEE, pp. 4631–4636.
Economic Commission for Europe, 2013. Inventory of most important bottlenecks and missing links in the E waterway network, Tech. Rep. ECE/TRANS/SC.3/159/Rev.1, Economic Commission for Europe, Inland Transport Committee, United Nations.
Farokhi, F., Shames, I., Johansson, K.H., 2014. Distributed mpc via dual decomposition and alternative direction method of multipliers. In: Maestre, J.M., Negenborn, R.R. (Eds.), Distributed Model Predictive Control Made Easy. Springer, pp. 115–131.
Fraichard, T., Asama, H., 2003. Inevitable collision states: a step towards safer robots?, in: Proceedings of International Conference on Intelligent Robots and Systems (IROS 2003), Vol. 1, pp. 388–393.
Hafner, M.R., Cunningham, D., Caminiti, L., Vecchio, D.D., 2013. Cooperative collision avoidance at intersections: algorithms and experiments. IEEE Trans. Intell. Transport. Syst. 14 (3), 1162–1175.
He, B.S., Yang, H., Wang, S.L., 2000. Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities. J. Optim. Theory Appl. 106 (2), 337–356.
Huang, Y., van Gelder, P., Wen, Y., 2018. Velocity obstacle algorithms for collision prevention at sea. Ocean Eng. 151, 308–321.
Ihle, I.A.F., 2006. Coordinated Control of Marine Craft, PhD Thesis.
Ihle, I.A.F., Jouffroy, J., Fossen, T.I., 2006. Formation control of marine surface craft: a lagrangian approach. IEEE J. Ocean. Eng. 31 (4), 922–934.
Keviczky, T., Borrelli, F., Fregene, K., Godbole, D., Balas, G.J., 2008. Decentralized receding horizon control and coordination of autonomous vehicle formations. IEEE Trans. Control Syst. Technol. 16 (1), 19–33.
Kujala, P., Hänninen, M., Arola, T., Ylitalo, J., 2009. Analysis of the marine traffic safety in the gulf of Finland. Reliab. Eng. Syst. Saf. 94 (8), 1349–1357.
Kuwata, Y., How, J.P., 2011. Cooperative distributed robust trajectory optimization using receding horizon milp. IEEE Trans. Control Syst. Technol. 19 (2), 423–431.
Li, G., Yamashita, A., Asama, H., Tamura, Y., 2012. An efficient improved artificial potential field based regression search method for robot path planning. In: Proceedings of IEEE International Conference on Mechatronics and Automation, Chengdu, China, pp. 1227–1232.
Li, S., Zheng, H., Negenborn, R.R., Lodewijks, G., 2015. Coordination for efficient transport over water. In: Proceedings of IEEE 19th International Conference on Computer Supported Cooperative Work in Design (CSCWD), pp. 389–394.
Li, S., Negenborn, R.R., Lodewijks, G., 2016. Distributed constraint optimization for addressing vessel rotation planning problems. Eng. Appl. Artif. Intell. 48, 159–172.
Liu, Y., Bucknall, R., 2015. Path planning algorithm for unmanned surface vehicle formations in a practical maritime environment. Ocean Eng. 97, 126–144.
Liu, Z., Zhang, Y., Yu, X., Yuan, C., 2016. Unmanned surface vehicles: an overview of developments and challenges. Annu. Rev. Control 41, 71–93.
Maestre, J.M., Negenborn, R.R. (Eds.), 2013. Distributed model predictive control made easy. Intelligent Systems, Control and Automation: Science and Engineering, vol. 69 Springer, Netherlands.
Mayne, D.Q., 2014. Model predictive control: recent developments and future promise. Automatica 50 (12), 2967–2986.
Monteil, J., Billot, R., Sau, J., Faouzi, N.E.E., 2014. Linear and weakly nonlinear stability analyses of cooperative car-following models. IEEE Trans. Intell. Transport. Syst. 15 (5), 2001–2013.
Muëller, M.A., Reble, M., Allgöwer, F., 2012. Cooperative control of dynamically decoupled systems via distributed model predictive control. Int. J. Robust Nonlinear Control 22 (12), 1376–1397.
Murray, R.M., 2007. Recent research in cooperative control of multivehicle systems. J. Dynam. Syst., Meas., Control 129 (5), 571–583.
Negenborn, R., Maestre, J., 2014. Distributed model predictive control: an overview and roadmap of future research opportunities. Control Syst. 34 (4), 87–97.
Olfati-Saber, R., Fax, J.A., Murray, R.M., 2007. Consensus and cooperation in networked multi-agent systems. Proc. IEEE 95 (1), 215–233.
Parthasarathi, R., Fraichard, T., 2007. An inevitable collision state-checker for a car-like vehicle. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 3068–3073.
Port of Rotterdam Authority, Copyright free port map high resolution, June 2017. <https://www.portofrotterdam.com/en/downloads/graphical-maps/port-of-rotterdam>.
Ren, W., Beard, R.W., 2008. Distributed Consensus in Multi-Vehicle Cooperative Control. Springer, London.
Richards, A., How, J.P., 2002. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In: Proceedings of 2002 American Control Conference, Vol. 3, pp. 1936–1941.
Richards, A., How, J., 2004. Decentralized model predictive control of cooperating uavs. In: Proceedings of 43rd IEEE Conference on Decision and Control, Vol. 4, pp. 4286–4291.
Rijkswaterstaat, Binnenvaartpolitiereglement, 1983. <http://wetten.overheid.nl/BWBR0003628/2017-01-01>.
Scattolini, R., 2009. Architectures for distributed and hierarchical model predictive control – a review. J. Process Control 19 (5), 723–731.
Shojaei, K., 2015. Leaderäfollower formation control of underactuated autonomous marine surface vehicles with limited torque. Ocean Eng. 105, 196–205.
Tam, C., Bucknall, R., 2013. Cooperative path planning algorithm for marine surface vessels. Ocean Eng. 57, 25–33.
Van den Berg, J., Lin, M., Manocha, D., 2008. Reciprocal velocity obstacles for real-time multi-agent navigation. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA 2008). IEEE, pp. 1928–1935.
Wang, S.L., Liao, L.Z., 2001. Decomposition method with a variable parameter for a class of monotone variational inequality problems. J. Optim. Theory Appl. 109 (2), 415–429.
Zhang, J., Zhang, D., Yan, X., Haugen, S., Soares, C.G., 2015. A distributed anti-collision decision support formulation in multi-ship encounter situations under COLREGs. Ocean Eng. 105, 336–348.
Zheng, H., Negenborn, R.R., Lodewijks, G., 2016. Fast ADMM for distributed model predictive control of cooperative waterborne AGVs. IEEE Trans. Control Syst. Technol. (99), 1–8.
Zheng, H., Negenborn, R.R., Lodewijks, G., 2016. Predictive path following with arrival time awareness for waterborne AGVs. Transport. Res. Part C: Emerg. Technol. 70, 214–237.
Zheng, H., Negenborn, R.R., Lodewijks, G., 2017. Robust distributed predictive control of waterborne AGVs–a cooperative and cost-effective approach. IEEE Trans. Cybernet. PP (99), 1–13.