

# Multivehicle Flocking With Collision Avoidance via Distributed Model Predictive Control

Yang Lyu<sup>1</sup>, Student Member, IEEE, Jinwen Hu<sup>2</sup>, Member, IEEE, Ben M. Chen, Member, IEEE, Chunhui Zhao, Fellow, IEEE, and Quan Pan, Member, IEEE

**Abstract**—Flocking control has been studied extensively along with the wide applications of multivehicle systems. In this article, the distributed flocking control strategy is studied for a network of autonomous vehicles with limited communication range. The main difference from the existing methods lies in that collision avoidance is considered a necessary condition while the vehicles are driven to follow a common desired trajectory under the proximity network. The sufficient conditions for system feasibility and stability are given by the proposed strategy. First, a centralized standard model predictive control (MPC) scheme is adopted to formulate the multivehicle flocking control problem by setting collision avoidance as an optimization constraint under the proximity network. Further, an equivalent distributed MPC (DMPC) is developed based on the consensus of local controllers under the existing framework of the alternating direction method of multiplier (ADMM). However, it may require infinite time to achieve consensus for all vehicles and, thus, the local controllers resulting in a limited number of ADMM iterations may not satisfy the given constraints. The constraints for each local controller are then modified so that the collision between vehicles is avoided all of the time. The feasibility and stability of the proposed method are analyzed under practical conditions. Simulation and experimental results show that the flocking of vehicles can track the common desired trajectory stably with no collisions by the proposed method.

**Index Terms**—Collision avoidance, flocking control, model predictive control (MPC), multivehicle system (MVS).

Manuscript received February 25, 2019; revised August 4, 2019; accepted September 21, 2019. Date of publication October 21, 2019; date of current version April 15, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61603303, Grant 6183309, and Grant 61703343, in part by the Natural Science Foundation of Shaanxi Province under Grant 2017JQ6005 and Grant 2017JM6027, in part by the China Postdoctoral Science Foundation under Grant 2018JQ6070610650, and in part by the Fundamental Research Funds for the Central Universities under Grant 3102018JCC003. This article was recommended by Associate Editor E. D. Sumer. (Corresponding author: Jinwen Hu.)

Y. Lyu is with the Key Laboratory of Information Fusion Technology, Ministry of Education, School of Automation, Northwestern Polytechnical University, Xi'an 710072, China, and also with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: lincoln1587@mail.nwpu.edu.cn).

J. Hu, C. Zhao, and Q. Pan are with the Key Laboratory of Information Fusion Technology, Ministry of Education, School of Automation, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: hujinwen@nwpu.edu.cn; zhaochunhui@nwpu.edu.cn; quanpan@nwpu.edu.cn).

B. M. Chen is with the Department of Mechanical and Automation Engineering, Chinese University of Hong Kong, Hong Kong (e-mail: bmchen@mae.cuhk.edu.hk).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TCYB.2019.2944892>.

Digital Object Identifier 10.1109/TCYB.2019.2944892

## I. INTRODUCTION

MULTIVEHICLE system (MVS) raises tremendous research interests in recent years [1], [2]. Compared with a single-vehicle system, the MVS usually has higher efficiency and operational capability in accomplishing complex tasks, such as transportation [3], search and rescue [4], and mapping [5]. As a prerequisite for MVS safe and autonomous operation, the collision-avoidance function has been recognized as one critical capability [6].

Flocking is a form of collective behavior of a large number of interacting agents with a common group objective which is characterized mainly by the three rules introduced by Reynolds [7]: 1) flocking centering; 2) collision avoidance; and 3) velocity matching. The flocking problem with collision avoidance has been studied under various settings. In [8], a theoretical framework is presented for flocking controller design and analysis, in which the collision avoidance for each vehicle with the flocking-mates and the obstacles is considered. The multiagent flocking problem with a virtual leader has been studied in [9] with a similar controller design method. In the aforementioned works, the collision avoidance is modeled as a collective potential function with regard to relative distance. A general collision-avoidance flocking control framework has been proposed in [10], which ensures collision avoidance under more general coupling forces. The flocking and connectivity preservation under proximity graphs have been studied in [11] and [12], and the collision avoidance is guaranteed simultaneously. More desired characteristics for the MVS system are considered in [13] and a bioinspired controller is proposed. The flocking and path following problems are studied and collision avoidance is guaranteed by designing a multifunctional control law in [14]. Besides the above controller design and analysis methods, a hybrid predator/intruder avoidance method for robot flocking, combined the reinforcement-learning-based decision process with a low-level flocking controller, has been proposed in [15]. A complete reinforcement-learning-based approach for UAV flocking is proposed in [16], where collision avoidance is incorporated into the reward function of the  $Q$ -learning scheme. Despite the methods above being able to handle the flocking control with collision avoidance, they may fail when more practical and combined constraints, such as the constraints on inputs and the states for collision avoidance, are imposed on the problem.

The model predictive control (MPC) scheme, with the feature of handling various constraints and optimized performance, is considered as a promising method to deal

with the flocking control problem. Compared with traditional controller design methods, the MPC-based method is supposed to have better control performance and a faster and smoother response. More important, the MPC-based method enables various constraints to be integrated into the problem explicitly. A centralized MPC for MVS has been proposed in [17], where the nonconvex collision-avoidance constraint is converted to a convex semidefinite program (SDP) problem using the Lagrangian relaxation method. One major drawback of the centralized method is that a data-processing center is needed to collect the data from each local vehicle and calculate the control command for each local controller. Compared with the centralized MPC method, the distributed MPC (DMPC) needs no data-processing center and uses peer-to-peer communication so that it is scalable for a large-scale network and robust against unexpected node or link failures [18]. The DMPC-based multiagent system control is widely studied under different conditions, such as the input constraint [19]–[21], communication constraint [22], communication delay [23], and so on. Works that further integrate the coupling factors, such as the collision avoidance between vehicles, into the cost function or constraints are described in [24] and [25]. One common technique to compute the local controller is to introduce the hypothetical information of neighbors so that consistency can be achieved in a distributed fashion. In the DMPC framework proposed in [26], each local controller also considers the hypothetical plan for its neighbors and all controllers are executed in a specific order rather than in parallel. A flocking control method with flock-mates collision avoidance is also presented in [27] based on the asynchronous DMPC. Although the asynchronous fashion is able to satisfy the coupling term by calculating the local strategy one by one, it lowers down the computational efficiency as only one local MPC is calculated at each sampling period. Correspondingly, the synchronous DMPC schemes adopted in [28] and [29] enable all controllers to calculate the control input during each sampling time. Specifically, [28] has presented a synchronous DMPC scheme to solve the tracking and formation problem with collision avoidance by using estimated information from neighbors. However, it fails to impose state and input constraints to the DMPC and, at the same time, guarantees recursive feasibility. Similar to [28], [29] has proposed a DMPC framework for formation control with both collision avoidance and obstacle avoidance capabilities. In addition, the input and state constraints have been incorporated into the DMPC framework and the recursive feasibility and closed-loop stability are guaranteed.

While the above works demonstrate the potential of DMPC in MVS, very few have been integrated into the flocking control problem with various practical constraints. The major challenge of implementing DMPC on the flocking task lies in that it is difficult to guarantee the system feasibility and stability when different constraints are imposed under practical conditions with a finite sampling period and prohibition of collision. Note that the works in [28] and [29] simplify the problem by defining a formation configuration rather than a common flocking reference trajectory so that the coupled collision-avoidance constraint can be satisfied. In this article,

a DMPC-ADMM flocking control framework is proposed to make an MVS track a desired trajectory under the constraints of finite communication range, collision avoidance, and bounded velocity and control input. First, a centralized standard MPC scheme is designed for the multivehicle flocking control by setting collision avoidance as an optimization constraint. Further, the centralized MPC is modified to a distributed fashion based on the consensus of local controllers under the framework of the alternating direction method of multipliers (ADMMs). However, it may require infinite time to achieve consensus for all vehicles and, thus, the local controllers resulting at the limited number of ADMM iterations may not satisfy the given constraints. The constraints for each local controller are then modified so that the collision between vehicles is avoided all of the time. The feasibility and stability of the proposed method are analyzed under practical conditions. Simulation and experimental results show that the flocking of vehicles can track the desired trajectory stably with no collisions by the proposed method.

The remainder of this article is organized as follows. Section II formulates the flocking control problem using the MPC scheme and Section III illustrates the DMPC-ADMM-based resolution to the problem formulated in Section II. Practical modifications of the DMPC-ADMM are described in Section IV to guarantee the feasibility and stability. An extension to the obstacle avoidance is discussed in Section V. Sections VI and VII give, respectively, the simulation and experimental validation for the proposed method and Section VIII concludes this article.

## II. PROBLEM FORMULATION

### A. Notations

The notations used in this article are fairly standard.  $\mathbb{R}^n$  denotes the  $n$ -dimensional Euclidean space.  $\mathbb{Z}$  and  $\mathbb{Z}^+$  denote the integer space and positive integer space, respectively. A matrix or vector transpose is denoted by a superscript  $\top$ . “ $\otimes$ ” denotes the Kronecker product between two matrices. The operator  $\|\cdot\|$  denotes the induced 2-norm of a matrix or the 2-norm of a vector. The operator  $\|\cdot\|_L$  is defined as  $\|\cdot\|_L = \sqrt{(\cdot)^\top L(\cdot)}$  with  $L$  being a positive-definite matrix, and the subscript is omitted if  $L$  being an identity matrix.  $\text{diag}(x)$  denotes a diagonal matrix with the vector  $x$  on its diagonal.  $\mathbf{1}_d$  is a column vector of size  $d$  with all elements to be 1.

### B. Dynamic Model

Consider a networked MVS consisting of  $n$  homogeneous members where each vehicle  $i$  is modeled as a second-order dynamic model

$$\begin{cases} \dot{\mathbf{q}}_i = \mathbf{p}_i \\ \dot{\mathbf{p}}_i = \mathbf{u}_i \end{cases} \quad (1)$$

with the 2-norm state and input constraints

$$\|\mathbf{p}_i\| \leq \bar{p}, \quad \|\mathbf{u}_i\| \leq \bar{u} \quad (2)$$

where  $\mathbf{q}_i, \mathbf{p}_i \in \mathbb{R}^m$  represent the position and velocity vectors of vehicle  $i$  in the  $m$ -dimensional global coordinate, respectively. In this article, we assume that each vehicle  $i$  is

fully aware of its own state in global coordinates, denoted as  $\mathbf{x}_i = [\mathbf{q}_i^\top, \mathbf{p}_i^\top]^\top \in \mathbb{R}^{2m}$ , through its navigation system (IMU, global positioning system (GPS) receiver, etc.). The control input, namely, the linear acceleration, is denoted as  $\mathbf{u}_i \in \mathbb{R}^m$ .  $\bar{p}$  and  $\bar{u}$  denote the scaler maximum of speed and acceleration, respectively. After an Euler discretization with step size  $T_s$ , the system dynamics is expressed as a linear model

$$\mathbf{x}_i(k+1) = A\mathbf{x}_i(k) + B\mathbf{u}_i(k) \quad (3)$$

where

$$A = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \otimes I_m \quad \text{and} \quad B = \begin{bmatrix} T_s^2/2 \\ T_s \end{bmatrix} \otimes I_m.$$

$k \in \mathbb{Z}^+$  denotes the sample time instance.

*Remark 1:* In this article, the control object is not restricted to a specific platform. Although the authors have not used any specific vehicle models, the method can be implemented on various vehicle platforms since there are inner-loop controllers that manipulate the physical actuators to achieve the effect of the desired input  $\mathbf{u}_i$ , where the proposed controller works as an outer-loop controller. Although the inertial dynamics is not considered directly, the physics constraints are integrated into the problem using velocity and control input constraints.

### C. Proximity Network

In this article, an undirected proximity graph  $\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k))$  is used to represent the communication topology of the MVS at each time instance  $k$ , where  $\mathcal{V}$  and  $\mathcal{E}(k) \in \mathcal{V} \times \mathcal{V}$  are, respectively, the set of vertices that stands for the local vehicles and the edge set that stands for the communication links. In the proximity network, the edge set  $\mathcal{E}(k)$  is defined according to the spatial distance between vehicles, namely,  $d_{ij}(k) = \|\mathbf{q}_i(k) - \mathbf{q}_j(k)\|$ , as

$$\mathcal{E}(k) = \{(i, j) | d_{ij}(k) < r_c, i, j \in \mathcal{V}, i \neq j\} \quad (4)$$

where  $r_c$  is the proximity network threshold. The neighborhood set of vehicle  $i$  is defined as  $N_i(k) \triangleq \{j | (i, j) \in \mathcal{E}(k)\}$ . In this article, we do not set any hypothesis on the connectivity of the network; therefore,  $N_i(k)$  maybe  $\emptyset$ , namely, vehicle  $i$  does not connect to any other vehicles.

### D. Flocking Control via MPC

The objective of our flocking controller is to track a common desired trajectory for multiple vehicles as a whole. To guarantee successful tracking of the common desired trajectory under the proximity network, all vehicles are assigned with the same desired trajectory. In this article, we assume that the time-varying reference state is subject to the linear dynamic model with constant velocity in a global coordinate, that is,  $\mathbf{x}_r(k+1) = A\mathbf{x}_r(k)$ . Besides tracking the same common reference trajectory, each vehicle should also alter its own trajectory according to the neighbor status so that a minimum separation distance between vehicles should be guaranteed to avoid collisions.

Based on the dynamic model and network model described above, the cost function within a prediction horizon  $H$  is

defined as

$$J(k) = \sum_{i \in \mathcal{V}} \left( \sum_{t=1}^{H-1} \left( \|\mathbf{x}_i(k+t|k) - \mathbf{x}_r(k+t)\|_Q^2 + \|\mathbf{u}_i(k+t-1|k)\|_R^2 \right) + \|\mathbf{x}_i(k+H|k) - \mathbf{x}_r(k+H)\|_P^2 \right) \quad (5)$$

where the expression  $(k+t|k)$  denotes the predictive time instance  $k+t$  from the state at  $k \in \mathbb{Z}^+$ . The first term on the right-hand side of (5) denotes the reference way-point tracking and reference velocity matching cost, and the second term is the predictive control cost. The third term denotes the terminal cost on state  $\mathbf{x}_i(k+H|k)$ .  $Q$ ,  $R$ , and  $P$ , respectively, denote the weight matrix of corresponding parts.

Collision avoidance is guaranteed by keeping a minimum separation distance between vehicles, that is

$$\|\mathbf{q}_i(k+t|k) - \mathbf{q}_j(k+t|k)\| \geq r_n, \quad i, j \in \mathcal{V} \quad (6)$$

within the prediction horizon  $t \in [1, H]$ .

*Problem 1:* Complying with the dynamic model (3) and the proximity network model (4), an MPC controller with a collision-avoidance constraint can be formulated as follows:

$$\min_{\mathbf{u}_i} J(k) \quad (7a)$$

$$\text{s.t. } \mathbf{x}_i(k+t+1|k) = A\mathbf{x}_i(k+t|k) + B\mathbf{u}_i(k+t) \quad (7b)$$

$$\|\mathbf{q}_i(k+t|k) - \mathbf{q}_j(k+t|k)\| \geq r_n \quad (7c)$$

$$\|\mathbf{p}_i(k+t|k)\| \leq \bar{p} \quad (7d)$$

$$\|\mathbf{u}_i(k+t-1|k)\| \leq \bar{u} \quad (7e)$$

$$\mathbf{u}_i(k+H-1) = \boldsymbol{\kappa}(\mathbf{x}_i(k+H-1)) \quad (7f)$$

$$i, j \in \mathcal{V}, i \neq j, t \in [1, H-1].$$

In Problem 1, (7a) is the MPC cost function, (7b) is the dynamic model of the vehicle  $i$ , (7c) denotes the collision-avoidance constraint between two vehicles, and (7d) and (7e) denote the state and input constraint of the vehicle, respectively. The terminal controller (7f) and terminal cost term  $\|\mathbf{x}_i(k+H|k) - \mathbf{x}_r(k+H)\|_P^2$  will guarantee the recursive feasibility and closed-loop stability of the controller. The derivation of the terminal design is detailed in Appendix A.

## III. DMPC-BASED FLOCKING WITH COLLISION AVOIDANCE

In this section, a DMPC-based flocking control scheme with collision avoidance is designed. Compared with the traditional centralized approach, which requires a fusion center to collect and compose a global state from every vehicle in the network in general, the distributed approach uses the peer-to-peer communication scheme that has the advantages of being scalable for a large-scale network and robust against unexpected node or link failures. In the DMPC scheme, each local controller  $i$  handles a subproblem involving vehicle  $i$  and its neighbors  $N_i$  and outputs the corresponding control sequence. In addition, to guarantee the consistency between local controllers, a global term is introduced and the global variables are updated using the consensus-based ADMM iterations.

### A. Subsystem MPC Formulation

First, we compose the state and control of the subsystem as  $\mathbf{X}_i = \{\mathbf{x}_{li}^\top\}_{l \in \{i\} \cup N_i}^\top$  and  $\mathbf{U}_i = \{\mathbf{u}_{li}^\top\}_{l \in \{i\} \cup N_i}^\top$ , where  $\mathbf{x}_{li}$  and  $\mathbf{u}_{li}$  denote the state and control input of vehicle  $l$  calculated in controller  $i$ . Without introducing ambiguity,  $\mathbf{x}_{ii}$  is written as  $\mathbf{x}_i$ . According to the dynamic model (3), we have the dynamics of the local system consisting of the neighbors of each agent as

$$\mathbf{X}_i(k+1|k) = \mathbf{A}_i \mathbf{X}_i(k) + \mathbf{B}_i \mathbf{U}_i(k) \quad (8)$$

with

$$\mathbf{A}_i = \mathbf{I}_{d_i} \otimes A, \quad \mathbf{B}_i = \mathbf{I}_{d_i} \otimes B$$

where  $\mathbf{I}_i$  is an identity matrix with size  $d_i$  and  $d_i = |N_i| + 1$ , and  $|N_i|$  is the number of neighbors of  $i$ .

Consequently, the cost function of each subsystem  $i$  can be rewritten in a compact form according to the cost function (5) as

$$J_i(k) = \sum_{t=1}^{H-1} \left( \|\mathbf{X}_i(k+t) - \mathbf{X}_r(k+t)\|_{Q_i}^2 + \|\mathbf{U}_i(k+t-1)\|_{R_i}^2 \right) + \|\mathbf{X}_i(k+H) - \mathbf{X}_r(k+H)\|_{P_i}^2 \quad (9)$$

where  $\mathbf{X}_r(k) \triangleq \mathbf{I}_{d_i} \otimes \mathbf{x}_r(k)$  is the predefined reference trajectory. The weight matrices  $Q_i$ ,  $R_i$ , and  $P_i$  are the distributed weight matrix with regard to  $Q$ ,  $R$ , and  $P$  defined, respectively, as  $L_i \otimes Q$ ,  $L_i \otimes R$ , and  $L_i \otimes P$ . The weight for the local vehicle is set as  $n-1$  and its neighbors as 1, that is,  $L_i = \text{diag}(n-1, 1, \dots, 1)$ .

### B. ADMM-Based Optimization

Assume that each local controller  $i$  is able to output a control sequence consisting of itself and the assumed control input of its neighbor. Apparently, there may be a deviation between the assumed control sequence and the actual control sequence from different distributed controllers, which may arouse conflicting behavior among vehicles. Therefore, it is necessary to adopt a negotiation scheme to guarantee the ‘‘opinions’’ of different controllers with regard to a common vehicle command to reach consensus. One optional approach is the consensus-based ADMM approach [30], [31], which is considered as a robust and decomposable framework dealing with the coupling distributed multiagent control problem.

In order to implement the consensus-based ADMM, first, we define the global vector  $\mathbf{Z}_i(k) = \{\mathbf{z}_j(k)\}_{j \in N_i(k) \cup \{i\}}^\top$ , where  $\mathbf{z}_j(k)$  is the average of vehicle  $j$ 's predicted states among its neighbors. The vector  $\mathbf{z}_j(k)$  is calculated as (23) in each local controller  $j$  and transferred to its neighbors  $N_j$ . Then, by introducing an equation constraint  $\mathbf{X}_i = \mathbf{Z}_i$  to each local controller  $i \in \mathcal{V}$ , the consensus can be reached among vehicle  $i$  and its neighbors  $j \in N_i$  as both  $\mathbf{Z}_i$  and  $\mathbf{Z}_{j \in N_i}$  share the same global component  $\mathbf{z}_i$ . The equation constraint is incorporated into (9) through the Lagrangian augmentation as

$$\begin{aligned} \bar{J}_i(k) = & J_i(k) + \sum_{t=1}^H \lambda_i^\top (\mathbf{X}_i(k+t|k) - \mathbf{Z}_i(k+t)) \\ & + \frac{\rho_i}{2} \|\mathbf{X}_i(k+t|k) - \mathbf{Z}_i(k+t)\|^2 \end{aligned} \quad (10)$$

where  $\lambda_i$  is the Lagrangian multiplier. The third term of (10) is an extra deviation penalty term with the weight factor  $\rho_i$ .

*Problem 2:* Based on the augmented cost function, we can describe the distributed controller  $i$  with regard to vehicle  $i$  and its neighbors  $N_i$  as  $\mathcal{P}_i$

$$\begin{aligned} \min_{\mathbf{U}_i(k)} \quad & \bar{J}_i(k) \\ \text{s.t.} \quad & \mathbf{X}_i(k+t+1|k) = \mathbf{A}_i \mathbf{X}_i(k+t|k) + \mathbf{B}_i \mathbf{U}_i(k+t) \\ & \|\mathbf{q}_i(k+t|k) - \mathbf{q}_j(k+t|k)\| \geq r_n \\ & \|\mathbf{p}_i(k+t|k)\| \leq \bar{p} \\ & \|\mathbf{p}_j(k+t|k)\| \leq \bar{p} \\ & \|\mathbf{u}_i(k+t-1|k)\| \leq \bar{u} \\ & \|\mathbf{u}_j(k+t-1|k)\| \leq \bar{u} \\ & \mathbf{u}_i(k+H-1|k) = \boldsymbol{\kappa}(\mathbf{x}_i(k+H-1|k)) \\ & \mathbf{u}_j(k+H-1|k) = \boldsymbol{\kappa}(\mathbf{x}_j(k+H-1|k)) \\ & j \in N_i(k), t \in [1, H]. \end{aligned}$$

The terminal design for the distributed controller is similar to the centralized detailed in Appendix A by just replacing the state and control of the entire flocking members with the subgroup member, for example,  $i \cup N_i$  in the local controller  $i$ .

Referring to the consensus-based ADMM method described in [31], each iteration here should include the ADMM parameters update and optimization solving of Problem 2. After a number of iterations, the consistency control input to each vehicle  $i \in \mathcal{V}$  can be obtained. Note that based on the ADMM update scheme, all DMPC subproblems  $\mathcal{P}_i, i \in \mathcal{V}$  can be solved in parallel. The ADMM iterations are detailed in Appendix B.

Finally, the discussed DMPC with the ADMM global variable update (DMPC-ADMM) is shown as Algorithm 1. During every sampling period, each distributed controller  $i \in \mathcal{V}$  solves a subproblem of the global problem in parallel, and the plan for  $\{i\} \cup N_i$  within the predictive horizon  $H$  is obtained. In each subproblem-solving process, ADMM iterates with the local optimization  $\mathcal{P}_i$  solving and global parameters update until the assumed plan and actual plan among different vehicles reach consensus. A final control input  $\mathbf{U}_i(k)$  is obtained after consensus is reached and the first element  $\mathbf{u}_i(k)$  is executed by vehicle  $i$ .

## IV. PRACTICAL MODIFICATION OF DMPC-ADMM ALGORITHM

The proposed DMPC-ADMM algorithm formulated in Section III is difficult to implement in practical applications as it may require infinite time to achieve consensus for all vehicles. To guarantee the DMPC performance as well as computational efficiency, a tradeoff is made by replacing the consensus condition with a bounded residual that can be achieved within a limited number of ADMM iterations as

$$\|\mathbf{x}_j(k) - \mathbf{z}_j(k)\| \leq \varepsilon, \quad j \in N_i \cup \{i\} \quad (12)$$

where  $\varepsilon$  is a predefined upper bound.

*Assumption 1:* The deviation between the local assumed plan and the global plan is much smaller than the predefined separation distance, that is,  $\varepsilon \ll r_n$ .

**Algorithm 1: DMPC-ADMM**


---

```

1 Initialize  $\mathbf{x}_{i \in \mathcal{V}}(k)$ ,  $k = 1$ ,
2 while Destination not reached do
3   for  $\forall i \in \mathcal{V}$  in parallel do
4     Initialize  $\boldsymbol{\lambda}_i = 0$ ,  $\mathbf{Z}_i = \mathbf{0}$ 
5     Receive  $\mathbf{x}_{j \in N_i}$  according to  $\mathcal{G}(k)$ 
6     repeat
7        $\mathbf{U}_i^*$ ,  $\mathbf{X}_i^* \leftarrow \mathcal{P}_i(\mathbf{X}_i(k|k), \boldsymbol{\lambda}_i, \mathbf{Z}_i)$ ,
8       Communicate  $\mathbf{x}_{ji}^*$  to  $j \in N_i$ ,
9        $\boldsymbol{\lambda}_i, \mathbf{Z} \leftarrow \text{ADMM\_Update}(\mathbf{X}_i^*, \boldsymbol{\lambda}_i, \mathbf{Z}_i)$ 
        according to (23) and (24)
10      until Consensus Reached;
11      Select the first control  $\mathbf{u}_i^*$  from  $\mathbf{U}_i^*$ 
12       $\mathbf{x}_i^*(k+1) = \mathbf{A}\mathbf{x}_i(k) + \mathbf{B}\mathbf{u}_i^*(k)$ .
13    end
14     $k \leftarrow k + 1$ 
15 end

```

---

Due to the residual, the feasibility and collision avoidance of the original consensus-based DMPC-ADMM method cannot be guaranteed. Thus, the controller of each vehicle has to be further refined in order to make the constraints satisfied.

*Lemma 1:* To guarantee the system feasibility and stability, the control input  $\mathbf{u}_{ji}$ , which denotes the control input with regard to vehicle  $j$  that is calculated in vehicle  $i$ 's DMPC controller, should be further tightened as

$$\begin{aligned} \|\mathbf{u}_{ji}(k|k)\| &\leq \bar{u} \\ \|\mathbf{u}_{ji}(k+1|k)\| &\leq \bar{u} - \eta_1 \\ \|\mathbf{u}_{ji}(k+t|k)\| &\leq \bar{u} - \eta_1 - \eta_2, t \in [2, H] \end{aligned} \quad (13)$$

where

$$\begin{aligned} \eta_1 &= 2\varepsilon \|\mathbf{I} \quad \mathbf{0}\| [\mathbf{A}\mathbf{B} \quad \mathbf{B}]^{-1} \mathbf{A}^2 \\ \eta_2 &= 2\varepsilon \|\mathbf{0} \quad \mathbf{I}\| [\mathbf{A}\mathbf{B} \quad \mathbf{B}]^{-1} \mathbf{A}^2 \end{aligned}$$

and the collision-avoidance constraint should be further relaxed as

$$\|\mathbf{q}_j(k+2|k+1) - \mathbf{q}_i(k+2|k+1)\| \geq d - \alpha, j \in N_i \quad (14)$$

with

$$\alpha = 2\varepsilon \left\| \left\{ \mathbf{I} - \mathbf{B}[\mathbf{I} \quad \mathbf{0}][\mathbf{A}\mathbf{B} \quad \mathbf{B}]^{-1} \mathbf{A} \right\} \mathbf{A} \right\|$$

if the stop criterion (12) can be reached.

*Proof:* See Appendix C. ■

*Remark 2:* According to Algorithm 1, the computation time of the proposed method is roughly proportional to the number of ADMM iterations during each sampling period, which is determined by the parameter  $\varepsilon$  defined in (12). In a practical scenario, a tradeoff should be made on selecting the  $\varepsilon$  to guarantee the above modification acceptable and to satisfy (12) simultaneously. Based on our observation, when the parameter is selected as  $\varepsilon = 0.01r_n$ , the stop criterion is very likely to be satisfied within ten ADMM iterations. It is obvious that the smaller the  $\varepsilon$  is set, the greater ADMM iterations that are needed to satisfy (12) and the more computation is needed.

*Assumption 2:* The communication range is large enough that satisfying  $r_c > 2(H+1)T_s\bar{v} + r_n$ .

The assumption indicates that in the worst case, there are  $H$  sample times before collision among the local vehicles' and its new neighbors. Therefore, collision avoidance is guaranteed even a vehicle does not consider the new neighbors in the next predict horizon  $H$ .

*Definition 1 (Lyapunov Stability [32]):* To analyze the system stability, define the following tracking error for the flock:

$$e(k) = \frac{1}{n} \sum_{i \in \mathcal{V}} \|\mathbf{x}_i(k) - \mathbf{x}_r(k)\|_Q. \quad (15)$$

Then, the system is said to be stable if and only if  $e(k)$  is uniformly bounded, that is, there exists a bound  $\delta > 0$  such that  $e(k) \leq \delta$  for all  $k > 0$ .

In this part, we would like to investigate the properties of the proposed flocking controller under practical condition (12), which is concluded as Theorem 1.

*Theorem 1:* By implementing the flocking control scheme in Algorithm 1, if the terminal condition (18)–(22) as well as the bounded residual (Assumptions 1 and 2) are satisfied, we have the following conclusions about the controller.

- 1) If all vehicles are able to find one feasible solution at time instance  $k$ , then for all subsequent time instance  $k' > k$ , there exist feasible solutions.
- 2) Collision between any vehicles is avoided, that is,  $\forall i, j \in \mathcal{V}, k > 0, \|\mathbf{q}_i(k) - \mathbf{q}_j(k)\| \geq r_n$ .
- 3) The closed-loop system is stable in the Lyapunov sense.

*Proof:* See Appendix D. ■

## V. EXTENSION TO OBSTACLE AVOIDANCE

Our proposed DMPC method is extended to the uncooperative obstacle avoidance problem in this section by modeling the obstacle avoidance constraint as

$$\|\mathbf{q}_i - \mathbf{q}_o\| \geq r_o \quad (16)$$

where  $\mathbf{q}_o$  is the obstacle position and  $r_o$  defines the minimum separation distance between the vehicle and the obstacle or the radius of the obstacle. The position of uncooperative obstacle is assumed to be known *a priori* or from the uncooperative localization methods. For the static obstacle, the position  $\mathbf{q}_o$  remains static within the predict horizon  $H$ , and for the moving obstacle, the position prediction is based on the uncooperative tracking results of the vehicle and assumed obstacle dynamic model as

$$\mathbf{x}_o(k+1|k) = f_o(\mathbf{x}(k)). \quad (17)$$

Different from the flock-mate collision avoidance, the obstacle collision-avoidance constraint is not coupled between vehicles, and the obstacle avoidance function can be realized by simply adding the obstacle avoidance constraint (16) and model prediction equation (17) into each subproblem  $\mathcal{P}_i$ . Although the obstacle avoidance is not theoretically included, we consider the flock-mates collision avoidance and obstacle avoidance simultaneously in both simulation and experiment, and the results show the effectiveness of both flock-mates collision avoidance and obstacle avoidance.

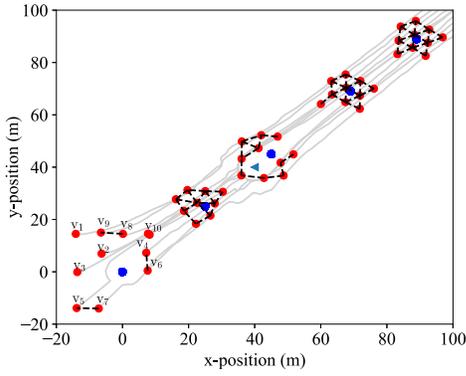


Fig. 1. Ten vehicles (labeled as  $v_1$ – $v_{10}$ ) flocking with collision avoidance and way-points tracking. The trajectories of ten vehicles are plotted in light gray. The vehicle, obstacle, and way-point are, respectively, represented as red balls, blue triangular, and blue balls. The dynamically changed network topology is plotted in the black dashed line.

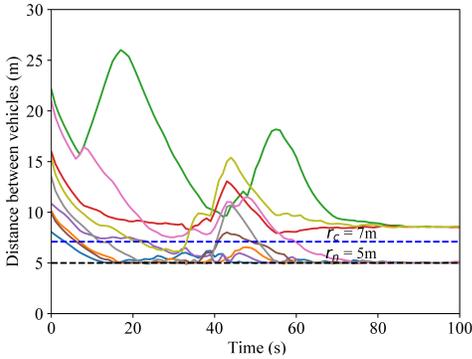


Fig. 2. Separation distances between vehicle  $v_2$  and other vehicles (solid line in different colors). The minimum separation distance and maximum communication distance between vehicles are  $r_c = 5$  m and  $r_n = 7$  m (dashed lines in black and blue).

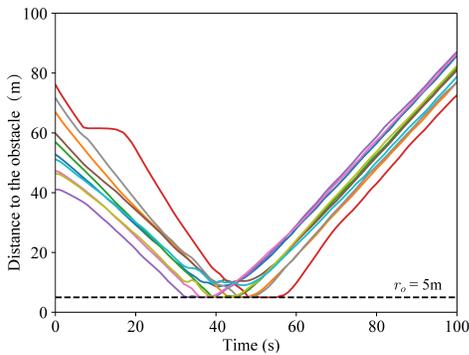


Fig. 3. Separation distances of different vehicles  $v_{1:10}$  with the obstacle (solid line in different colors). The minimum separation distance is  $r_o = 5$  m (dashed line in black).

## VI. SIMULATIONS

In this section, the effectiveness of the DMPC-ADMM algorithm is illustrated using simulation with synthetic 2-D data. The DMPC-ADMM is implemented based on the cvxpy toolbox [33] and the QCQP solver [34].

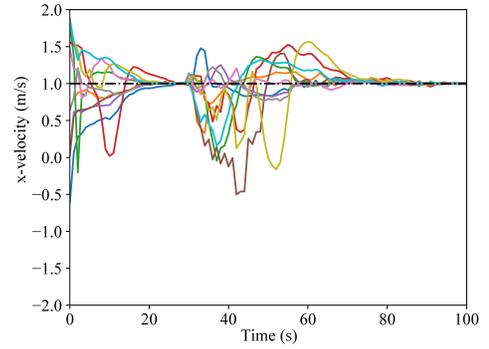


Fig. 4. Velocities of vehicles  $v_{1:10}$  (solid lines in different colors). The velocity reference along the  $x$ -axis is  $v_x = 1$  (black dashed line).

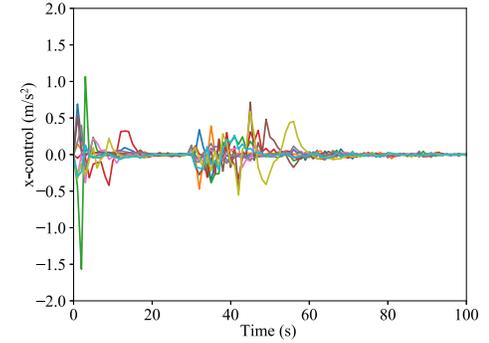


Fig. 5. Control inputs in the  $x$ -axis of vehicles  $v_{1:10}$  (solid lines in different colors).

### A. MVS Collision Avoidance

In 2-D simulation, the scenario includes ten mobile vehicles which are assumed to have a homogeneous linear model (3) and one static obstacle in a plane. The velocity and control constraints are set, respectively, as  $\bar{v} = 3$  and  $\bar{u} = 2$ . The minimum separation distances for flock mates and obstacle are set as  $r_n = r_o = 5$ , and the maximum communication range between flock-mates is set as  $r_c = 7$ . The stop criterion of the ADMM iteration is defined as either reaching ten iterations or satisfying that  $\|\mathbf{x}_i - \mathbf{z}_i\| \leq \varepsilon = 0.1$ . The ten mobile vehicles are randomly placed around the point  $[0, 0]^T$  at the same time satisfying the collision-avoidance constraints. The dynamic reference for the flocking is initialized as  $\mathbf{x}_r(0) = [0, 0, 1, 1]^T$  and the static obstacle is initialized as  $\mathbf{x}_o(0) = [40, 40, 0, 0]^T$ .

The simulation results are shown in Figs. 1–5 to illustrate the effectiveness of the DMPC-ADMM algorithms on flocking control with flock-mates and obstacle collision avoidance. The trajectories of the flocking with vehicles 1–10 are plotted in Fig. 1 with five representative snap-shots showing the flocking evolution during obstacle avoidance. The communication (the black dashed line) dynamically changes during the flocking process without a control center. The separation distances between vehicles are plotted in Fig. 2 with the predefined threshold  $r_n$  in the black dashed line. Similarly, the relative distances between vehicles and the static obstacle are shown in Fig. 3 with the predefined threshold  $r_o$  in the black dashed line. The above two figures show the collision-avoidance capabilities with regard to flock-mates and obstacles, respectively.

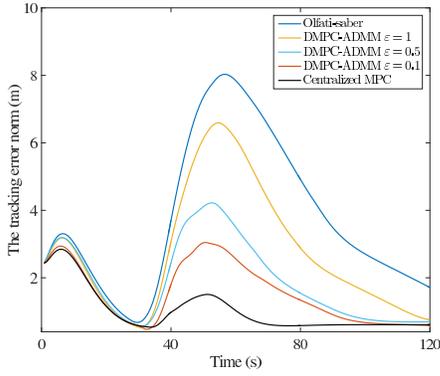


Fig. 6. Tracking error  $\bar{e}(k)$  of flocking based on DMPC with different  $\varepsilon$  versus centralized MPC.

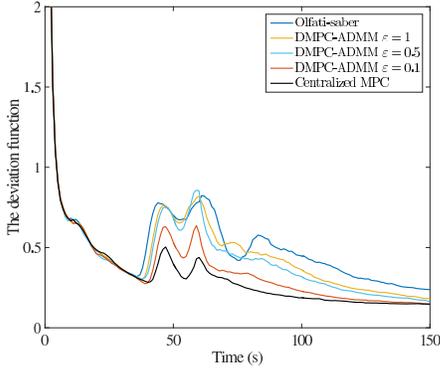


Fig. 7. Deviation function  $v(k)$  based on DMPC with different  $\varepsilon$  versus centralized MPC.

The velocity and control input are plotted in Figs. 4 and 5, respectively, which indicate that the constraints on velocity and control input are satisfied.

### B. Comparison of Centralized MPC

In this scenario, to further illustrate the advantage of the proposed DMPC-ADMM method, more detailed comparisons between the DMPC algorithm, the centralized MPC methods [17], and the method proposed by Olfati-Saber [8] are given based on the Monte Carlo simulations. Specifically, we select the stop criterion as  $\varepsilon \in \{0.1, 0.5, 1\}$  for the DMPC method. For each possible  $\varepsilon$ , a set of 100 Monte Carlo simulations is carried out which has a similar setup to the one trial simulation above. In order to perform the comparison, we define the following three flocking control indices.

- 1) the flocking reference tracking error which is defined in (15);
- 2) the deviation energy represents a flock number different from the  $\alpha$ -lattice, which is considered as the optimized flocking configuration [8]

$$v(k) = \frac{1}{|\mathcal{E}(k)| + 1} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i} (\|q_i(k) - q_j(k)\| - r_n)^2;$$

- 3) the proximity network edge number  $|\mathcal{E}(k)|$ .

The Monte Carlo simulation results of the aforementioned three indices of the proposed DMPC method versus the centralized MPC and the method in [8] are shown in Figs. 6–8.

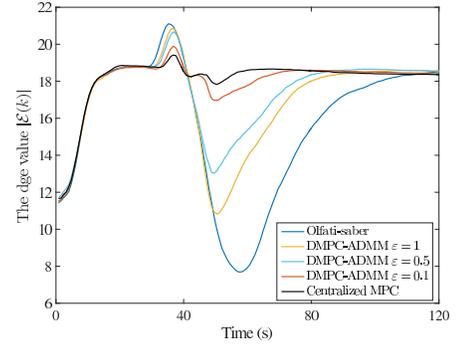


Fig. 8. Communication link number  $|\mathcal{E}(k)|$  based on DMPC with different  $\varepsilon$  versus centralized MPC.

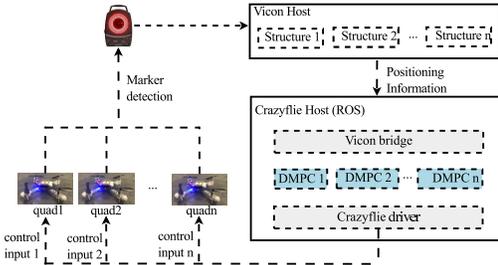


Fig. 9. Information flow of the experimental setup.

It is obvious that the three methods are able to realize similar flocking behavior, and the centralized method is able to realize the best performance. The smaller  $\varepsilon$  is set, the closer the performance that the centralized MPC can be achieved by the DMPC-ADMM method. Nevertheless, to obtain results under more strict criteria, more ADMM iterations are required, which will consume more computation and communication resources. Compared with the method proposed in [8], the proposed method is able to realize faster response and better-flocking behavior and trajectory tracking.

## VII. EXPERIMENT

In this part, our proposed DMPC-ADMM algorithm is testified by the experiment with five Crazyflie 2.0 quadrotors [35] in the indoor environment. Four quadrotors are driven to flock by the proposed algorithm while one quadrotor is used as a moving obstacle with constant velocity. The experimental setup is shown in Fig. 9. The host computer is used mainly for the complicated computation of the controller because the quadrotors used in the experiments do not have sufficient computation capability to solve the desired controller in real time. In addition, the experiments are implemented indoor, and the vehicle positions are obtained using the Vicon [36] positioning system which requires the host computer to translate signals. Nevertheless, the controllers of different vehicles are computed in parallel and distributively by the host computer. Four DMPC nodes under the robotics operation system (ROS) are established to control the four quadrotors in parallel. The communication is realized by the publishing/subscribing scheme between nodes according to the dynamic graph. During the experiment, the four Crazyflies (labeled as quad1, quad2, quad3, and quad4)

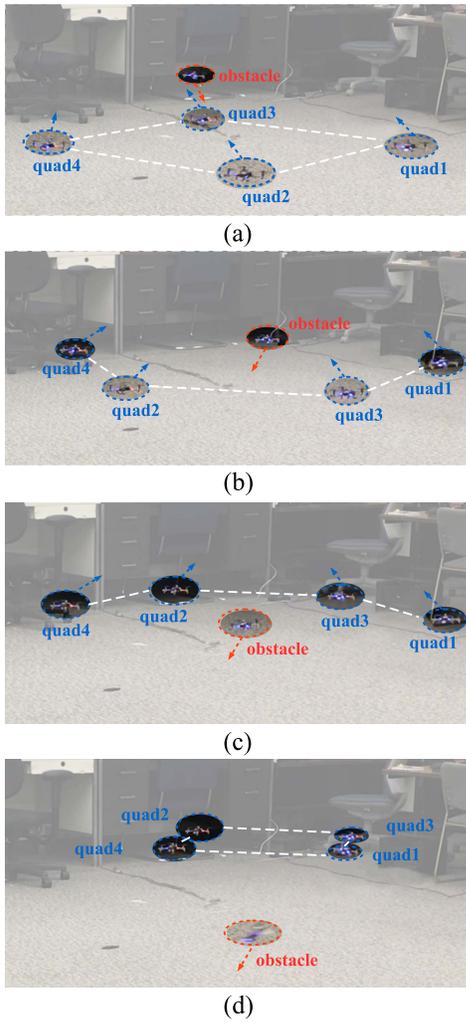


Fig. 10. Snapshots of the flocking collision avoidance experiments: (a) initialization flocking, (b) and (c) collision avoidance, and (d) flocking reforming, where the flock members are marked by the blue circles, the dynamic obstacle is marked by the red circle, and the communication is marked by the white dashed line.

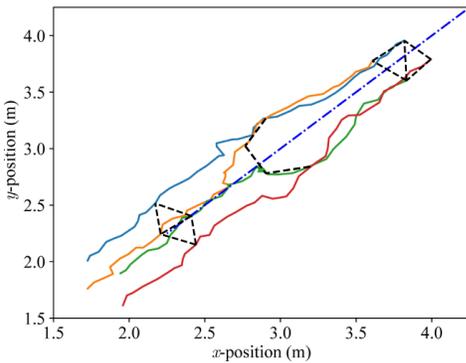


Fig. 11. Trajectories of four vehicles in different colors, where the obstacle trajectory is represented by the blue dashed line and the communication links are represented by the black dashed lines.

are randomly placed in the area  $\{(x, y)|x, y \in [1.5, 2]\}$  with the same height 0.5 in the Vicon coordinate. The dynamic constraints are set as  $\bar{v} = 0.3$  and  $\bar{u} = 0.2$ , respectively. The reference state for the flocking is initialized as  $\mathbf{x}_r(0) =$

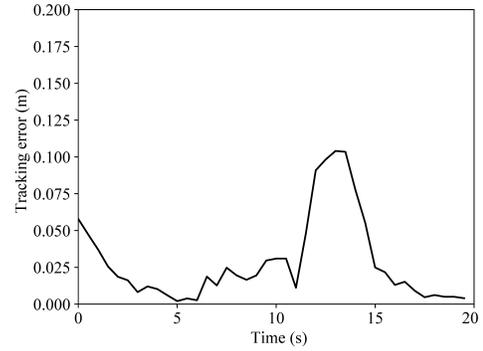


Fig. 12. Tracking error.

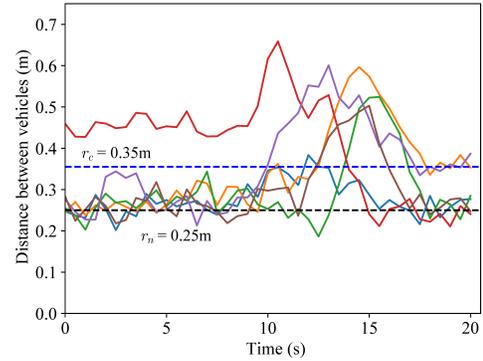


Fig. 13. Separation distance between different vehicles with  $r_c = 0.35$  and  $r_n = 0.25$ .

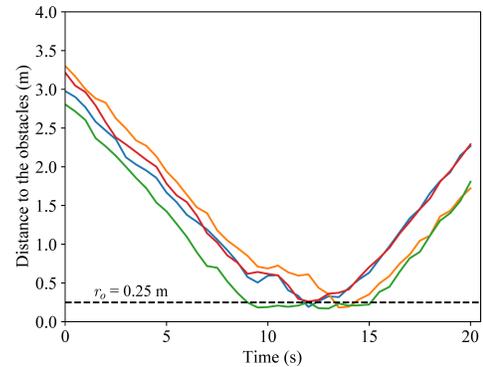


Fig. 14. Separation distance between each vehicle and the obstacle with  $r_o = 0.25$ .

$[2.0, 2.0, 0.1, 0.1]^T$ . The moving quadrotors are initialized as  $\mathbf{x}_o = [4.5, 4.5, -0.1, -0.1]^T$ . In this head-on scenario, the collision-avoidance constraints are set as  $r_n = r_o = 0.25$  and the maximum communication range is set as  $r_c = 0.35$ . All of the above experiment is set up in metric.

The experiment is demonstrated using four snapshots as Fig. 10, and the related trajectories of the flocking and obstacle are plotted in Fig. 11 which demonstrates both the flocking forming and collision avoidance by our proposed method. Especially, the reference waypoints tracking error  $e(k)$  is shown in Fig. 11 which indicates that the tracking is within a very low error level except during obstacle avoidance. The collision-avoidance performance with regard to flock-mates and obstacles is plotted, respectively, in Figs. 13 and 14.

## VIII. CONCLUSION

In this article, a distributed flocking control strategy for networked autonomous vehicles with a limited communication range was proposed. As the main contribution, the proposed strategy allows all vehicles to follow a desired trajectory and avoid collisions. The sufficient conditions for the system feasibility and stability were provided. Simulation and experimental results showed that the team of vehicles can track the desired trajectory stably with no collisions by the proposed method.

In future work, we will further add on the requirement for team formations. In most applications of formation control, accurate path planning is carried out in advance of motion control, and the initial positions of vehicles are always placed carefully, which consumes a lot of time in a large network. Inappropriate design of motion control may not lead to desired team formation, though it can help vehicles avoid collision. Therefore, new collaboration schemes will be developed for team formation control without such limitations.

 APPENDIX A  
 TERMINAL DESIGN

*Terminal Controller:* To guarantee system stability, a terminal controller and a terminal cost function with regard to Problem 1 are designed as follows. First, we define the terminal controller for each local controller  $i$  as

$$\boldsymbol{\kappa}_i = K\mathbf{x}_i(k+H-1) + G\mathbf{x}_r(k+H-1).$$

Substitute the control input in the dynamic (3) with the terminal controller  $\boldsymbol{\kappa}_i(\mathbf{x}(k+H-1))$  as

$$\begin{aligned} \mathbf{x}_i(k+H) &= A\mathbf{x}_i(k+H-1) + B(K\mathbf{x}_i(k+H-1) \\ &\quad + G\mathbf{x}_r(k+H-1)) \end{aligned}$$

and subtracting  $\mathbf{x}_r(k+H) = A\mathbf{x}_r(k+H-1)$  on both sides, we finally obtain

$$\begin{aligned} \mathbf{x}_i(k+H) - \mathbf{x}_r(k+H) &= (A+BK)(\mathbf{x}_i(k+H-1) - \mathbf{x}_r(k+H-1)) \\ &\quad + B(K+G)\mathbf{x}_r(k+H-1). \end{aligned} \quad (18)$$

For the reference state tracking purpose, namely, to cancel the error between state and reference state, the terminal controller should satisfy the following conditions:

- 1)  $A+BK$  is stabilized;
- 2)  $B(K+G)\mathbf{x}_r(k) = \mathbf{0}$ .

*Terminal Cost:* To guarantee the stability, the following equation should be satisfied [37]:

$$\begin{aligned} \|\mathbf{x}_i(t) - \mathbf{x}_r(t)\|_Q^2 + \|K\mathbf{x}_i(t) + G\mathbf{x}_r(t)\|_R^2 \\ - \|\mathbf{x}_i(t) - \mathbf{x}_r(t)\|_P^2 + \|\mathbf{x}_i(t+1) - \mathbf{x}_r(t+1)\|_P^2 \leq 0. \end{aligned} \quad (19)$$

Substituting the terminal controller  $\boldsymbol{\kappa}(\mathbf{x}_i(k+H))$  into the left-hand side of (19) and using the triangulation inequation, we have

$$\begin{aligned} \|\mathbf{x}_i(t) - \mathbf{x}_r(t)\|_Q^2 + \|K\mathbf{x}_i(t) + G\mathbf{x}_r(t)\|_R^2 - \|\mathbf{x}_i(t) - \mathbf{x}_r(t)\|_P^2 \\ + \|\mathbf{x}_i(t+1) - \mathbf{x}_r(t+1)\|_P^2 \end{aligned}$$

$$\begin{aligned} &= \|\Delta\mathbf{x}_i(t)\|_Q^2 + \|(K+G)\mathbf{x}_r(t) - K\Delta\mathbf{x}_i(t)\|_R^2 + \|\Delta\mathbf{x}_i(t+1)\|_P^2 \\ &\quad - \|\Delta\mathbf{x}_i(t)\|_P^2 \\ &\leq \|\Delta\mathbf{x}_i(t)\|_Q^2 + 2\left(\|\Delta\mathbf{x}_i(t)\|_{K^\top RK}^2 + \|\mathbf{x}_r(t)\|_{(K+G)^\top R(K+G)}\right) \\ &\quad + \|\Delta\mathbf{x}_i(t)\|_{(A+BK)^\top P(A+BK)} - \|\Delta\mathbf{x}_i(t)\|_P^2 \\ &= \|\Delta\mathbf{x}_i(t)\|_\Xi^2 + \|\mathbf{x}_r(t)\|_\Upsilon^2 \end{aligned} \quad (20)$$

where  $\Delta\mathbf{x}_i = \mathbf{x}_i - \mathbf{x}_r, t = H+k$ , and

$$\begin{aligned} \Xi &= Q + 2K^\top RK + (A+BK)^\top P(A+BK) - P \\ \Upsilon &= (K+G)^\top (K+G). \end{aligned}$$

Given the terminal controller condition 2), the final controller  $\|\mathbf{x}_r\|_\Upsilon = 0$ ; thus, inequality (19) can be further simplified as

$$\|\Delta\mathbf{x}_i\|_\Xi^2 \leq 0. \quad (21)$$

To guarantee (21),  $P$  should be selected, satisfying

$$Q + 2K^\top RK + (A+BK)^\top P(A+BK) - P \succcurlyeq 0.$$

Based on the above terminal design, there is no coupling in each local controller's terminal design; therefore, the terminal design process of Problem 2 is the same as the centralized terminal design method as above.

 APPENDIX B  
 ADMM ITERATIONS

Based on the distributed controller described in Problem 2, the ADMM is iterated as

$$\mathbf{X}_i^{(k+1)} = \mathcal{P}_i\left(\mathbf{X}_i, \mathbf{U}_i^{(k)}, \mathbf{Z}_i^{(k)}, \boldsymbol{\lambda}_i^{(k)}\right) \quad (22)$$

$$\mathbf{z}_i^{(k+1)} = \frac{1}{|N_i|+1} \sum_{j \in N_i \cup i} \mathbf{x}_{ij}^{(k+1)} \quad (23)$$

$$\boldsymbol{\lambda}_i^{(k+1)} = \boldsymbol{\lambda}_i^{(k)} + \rho_i\left(\mathbf{X}_i^{(k+1)} - \mathbf{Z}_i^{(k+1)}\right) \quad (24)$$

where the superscribe  $k'$  denotes the ADMM iteration number, and  $\mathbf{x}_{ij}$  denotes the assumed predicted state with regard to vehicle  $i$  from controller  $j$ .

 APPENDIX C  
 PROOF OF LEMMA 1

Assume that after we replace the *consensus reached* with (12) in Algorithm 1 at time instance  $k$ , there is a solution for  $\mathcal{P}_i, i \in \mathcal{V}$  with predictive control input and the state denoted, respectively, as  $\mathbf{u}_i^*(k+t-1|k)$  and  $\mathbf{x}_i^*(k+t|k)$ ,  $t \in [1, H]$ . By implementing the control input  $\mathbf{u}_i^*(k|k)$ , vehicle  $i$  reaches a new state  $\mathbf{x}_i^*(k+1)$ . At the time instance  $k+1$ , the deviation between the vehicle state and the assumed state from its neighbors  $\mathbf{x}_{ij}^*(k+1|k)$  is

$$\begin{aligned} \|\mathbf{x}_i^*(k+1) - \mathbf{x}_{ij}^*(k+1|k)\| &= \|\mathbf{x}_i^*(k+1|k) - \mathbf{z}_i(k+1) + \mathbf{z}_i(k+1) - \mathbf{x}_{ij}^*(k+1|k)\| \\ &\leq \|\mathbf{x}_i^*(k+1) - \mathbf{z}_i(k+1)\| + \|\mathbf{x}_{ij}^*(k+1|k) - \mathbf{z}_i(k+1)\| \\ &\leq 2\varepsilon. \end{aligned} \quad (25)$$

Consequently, we can express  $\mathbf{x}^*(k+t)$  using  $\mathbf{x}_{ij}^*(k+t|k), j \in N_i$  as

$$\begin{aligned} \mathbf{x}_i^*(k+1|k) &= \mathbf{x}_{ij}^*(k+1|k) + \boldsymbol{\xi} \\ &= A\mathbf{x}_{ij}^*(k|k) + B\mathbf{u}_{ij}^*(k) + \boldsymbol{\xi}, \quad \|\boldsymbol{\xi}\| \leq 2\varepsilon. \end{aligned} \quad (26)$$

From (26), the actual plan of a vehicle can be treated as the assumed plan from its neighbors with a bounded noise  $\boldsymbol{\xi}$ . By implementing the two-step robust feasible method [38], we can force a feasible solution  $\mathbf{x}'(k+3|k+1)$  to rejoin the prior results at  $(k+3|k)$ , namely

$$\mathbf{x}'_i(k+3|k+1) = \mathbf{x}_{ij}^*(k+3|k). \quad (27)$$

Combined with the second-order dynamic model (3), the two-step control input is designed as

$$\begin{aligned} \mathbf{u}'_{ji}(k+1|k+1) &= \mathbf{u}_{ji}^*(k+1|k) - [\mathbf{I} \quad \mathbf{0}][AB \quad B]^{-1}A^2\boldsymbol{\xi} \\ \mathbf{u}'_{ji}(k+2|k+1) &= \mathbf{u}_{ji}^*(k+2|k) - [\mathbf{0} \quad \mathbf{I}][AB \quad B]^{-1}A^2\boldsymbol{\xi}. \end{aligned} \quad (28)$$

*Remark 3:* The inverse of the matrix  $[AB \quad B]$  is guaranteed on the second-order system with the status including position  $q \in \mathbb{R}^m$  and velocity  $p \in \mathbb{R}^m$ , and the control input acceleration  $u \in \mathbb{R}^m$ , which means  $A \in \mathbb{R}^{2m \times 2m}$  and  $B \in \mathbb{R}^{2m \times m}$ .

Then, the control input can be further tightened as

$$\begin{aligned} \|\mathbf{u}_{ji}(k|k)\| &\leq \bar{u} \\ \|\mathbf{u}_{ji}(k+1|k)\| &\leq \bar{u} - \eta_1 \\ \|\mathbf{u}_{ji}(k+t|k)\| &\leq \bar{u} - \eta_1 - \eta_2, \quad t \in [2, H] \end{aligned} \quad (29)$$

where

$$\begin{aligned} \eta_1 &= \|\mathbf{I} \quad \mathbf{0}\| [AB \quad B]^{-1}A^2\| \cdot 2\varepsilon \\ \eta_2 &= \|\mathbf{0} \quad \mathbf{I}\| [AB \quad B]^{-1}A^2\| \cdot 2\varepsilon. \end{aligned}$$

By implementing the feasible control input (13), the transition state  $\mathbf{x}'_i(k+2|k+1)$  is

$$\begin{aligned} \mathbf{x}'_i(k+2|k+1) &= \mathbf{x}_{ij}^*(k+2|k) \\ &+ \left\{ \mathbf{I} - B[\mathbf{I} \quad \mathbf{0}][AB \quad B]^{-1}A \right\} A\boldsymbol{\xi}. \end{aligned} \quad (30)$$

To ensure the feasibility of the problem at time instance  $k+2$ , we further relax the collision-avoidance constraint as

$$\|\mathbf{q}_j(k+2|k+1) - \mathbf{q}_i(k+2|k+1)\| \geq d - \alpha, \quad j \in N_i \quad (31)$$

with

$$\alpha = \left\| \left\{ \mathbf{I} - B[\mathbf{I} \quad \mathbf{0}][AB \quad B]^{-1}A \right\} A \right\| \cdot 2\varepsilon. \quad (32)$$

Based on the modified flock-mate collision-avoidance constraints and control input constraints, the new solution at  $k+1$  can rejoin the resolution at  $k$  on predictive time instance  $k+3$ . Then, the feasibility is proved for the unchanged topology.

## APPENDIX D PROOF OF THEOREM 1

1) We assume that there is no topology change during time instance  $k$  to  $k+1$ . Then, a feasible solution for vehicle  $i \in \mathcal{V}$ , which includes both control input of  $i$  and its neighbors  $j \in N_i$ , needs to be found. Assume that at time instance  $k+1$ , each vehicle  $j \in N_i$  reaches a new state  $\mathbf{x}_j^*(k+1)$  by executing its local control input  $\mathbf{u}_j^*(k)$ . Due to the possible difference between  $\mathbf{x}_j^*(k+1)$  and  $\mathbf{x}_{ji}^*(k+1)$ , the subsequent solution of controller  $i$  with regard to  $j$ , namely,  $\mathbf{u}_{ji}^*(k+t|k), t \in [1, H]$  is not applicable. According to Lemma 1, the two-step feasible control input (28) can be used, and the state of vehicle  $j$  is able to rejoin the previous plan on agent  $i$ , namely,  $\mathbf{x}'_{ji}(k+3|k+1) = \mathbf{x}_{ji}^*(k+3|k)$ . Then, the subsequent control input  $\mathbf{u}'_{ji}(k+t|k+1) = \mathbf{u}_{ji}^*(k+t|k)|_{t=3:H-1}$  can be treated as a feasible solution. In addition, by implementing the terminal controller (18), the way-point tracking norm  $\|\mathbf{X}_i(k+H) - \mathbf{X}_r(k+H)\|_{P_i}^2$  is monotonously nonincreasing which means the terminal set is a positively invariant set that satisfies the flock-mate constraints. In the local controller  $i$ , a feasible solution for each neighbor  $j$  at time  $k+1$  can be concluded as

$$\left\{ \mathbf{u}'_{ji}(k+1|k+1), \mathbf{u}'_{ji}(k+2|k+1), \left\{ \mathbf{u}_{ji}^*(l|k) \right\}_{l \in k+3:k+H} \right\}. \quad (33)$$

Furthermore, we consider the case that the topology changes between time instance  $k$  and  $k+1$ . Apparently, for each local problem  $\mathcal{P}_i$  formulation, the topology change could be decomposed as two parts, that is, removing the existing neighbors, denoted as  $l_i^-$ , and adding new neighbors, denoted as  $l_i^+$ . Then, a feasible solution of  $\mathcal{P}_i$  at time instance  $k+1$  can be found based on (33) by removing the control input  $\mathbf{u}'_{l_i^-}$  and adding the new neighbors' control input  $\mathbf{u}_{l_i^+} = \mathbf{u}_{l_i^+}^*$  which is calculated by vehicle  $l_i^+$  in time instance  $k$ . Although the collision avoidance between  $i$  and  $l_i^+$  is not considered in calculation of the control input  $\mathbf{u}_{l_i^+}$  and  $\mathbf{u}_i^*$ , according to Assumption 2, collision avoidance is still guaranteed during the prediction horizon  $[k+1, k+H+1]$ .

Conclusively, a feasible solution at  $k+1$  could always be found whether the topology is changed, and recursively the feasibility is proved.

2) The collision avoidance is modeled as constraints between each vehicle and its flockmates in Problem 2, and are naturally satisfied by the feasible solution (33). The collision-free trajectories for all vehicles are guaranteed.

3) Let  $V_i(k)$  denote the part of the MPC cost function (5) that is related to the status of vehicle  $i$ , and further let  $\mathbf{V} \triangleq [V_1, \dots, V_i, \dots, V_n]$  denote the cost set of the overall system. As a result, the cost function of each subproblem and the centralized problem can, respectively, be expressed as  $J_i = (n-1)V_i + \sum_{l \in N_i} V_l$  and  $J = \sum_{i \in \mathcal{V}} V_i$ . Note that the function  $J_i$  is discontinuous due to the possible topology change but  $J$  is continuous since it is dependent on the status and control input of all vehicles. Denoting the set of all subproblems cost function as  $\mathbf{J} \triangleq [J_1, \dots, J_i, \dots, J_n]^T$ ,  $\mathbf{J}(k)$  can be expressed as

$$\mathbf{J}(k) = D(k)\mathbf{V}(k) \quad (34)$$

where  $D(k) = (n-1)\mathbf{I}_{n \times n} + \mathcal{A}(k)$ ,  $\mathcal{A}(k)$  denotes the adjacency matrix of the proximity network at time instance  $k$ .  $D(k)$  is a diagonal dominate matrix and is therefore invertible. Defining a row vector  $\zeta(k) = \mathbf{1}_{1 \times n} D(k)^{-1}$ , it is provable that all elements of  $\zeta(k)$ ,  $k > 0$  are non-negative based on the Cramer rule. By multiplying both sides of (34) with  $\zeta(k)$ , the centralized cost function can be expressed with the subsystem cost functions linearly as

$$J(k) = \zeta(k)\mathbf{J}(k). \quad (35)$$

Next, we will prove that the centralized cost function is nonincreasing, that is,  $J(k+1) \leq J(k)$  based on the proposed DMPC-ADMM controller. Denote  $J_i^*(k)$  and  $J_i'(k)$  as the cost value of each controller  $i$  based, respectively, on the calculated results from Algorithm 1 and feasible solution (33). The cost value of specific time instance  $k+t$ ,  $t \in [1, H]$  in  $J_i^*(k)$  and  $J_i'(k)$  are denoted, respectively, as  $J_i^*(k+t|k)$  and  $J_i'(k+t|k)$ , that is,  $J_i^*(k) = \sum_{t=1}^H J_i^*(k+t|k)$  and  $J_i'(k) = \sum_{t=1}^H J_i'(k+t|k)$ .

First, we assume that there is no topology change between time instance  $k$  and  $k+1$ , then the difference between the cost on the feasible solution at  $k+1$  and the previous obtained solution at  $k$  can be obtained as

$$\begin{aligned} & J_i'(k+1) - J_i^*(k) \\ &= \sum_{t=1}^H J_i'(k+t+1|k+1) - \sum_{t=1}^H J_i^*(k+t|k) \\ &= J_i'(k+2|k+1) + \sum_{t=2}^{H-2} J_i'(k+t+1|k+1) \\ &\quad + \sum_{t=H-1}^H J_i'(k+t+1|k+1) - \sum_{t=1}^2 J_i^*(k+t|k) \\ &\quad - \sum_{t=3}^{H-1} J_i^*(k+t|k) - J_i^*(k+H|k). \end{aligned} \quad (36)$$

According to the terminal design in Section III-B and the definition of feasible solution in (33), the following equations can be obtained, respectively, as:

$$\begin{aligned} & \sum_{t=H-1}^H J_i'(k+t+1|k+1) - J_i^*(k+H|k) \\ & \leq \sum_{t=H-1}^H J_i^*(k+t+1|k+1) - J_i^*(k+H|k) \leq 0 \end{aligned}$$

and

$$\sum_{t=3}^{H-2} J_i'(k+t+1|t+1) - \sum_{t=4}^{H-1} J_i^*(k+t|k) = 0.$$

The remaining part of the right-hand side of (36) is  $\sum_{t=1}^2 (J_i'(k+t+1|k+1) - \sum_{t=1}^3 J_i^*(k+t|k))$ . Based on Assumption 1, the feasible control input (33) and transition state (30)

$$\sum_{t=1}^2 \left( J_i'(k+t+1|k+1) - \sum_{t=1}^3 J_i^*(k+t|k) \right) \leq 0.$$

Finally

$$J_i^*(k+1) \leq J_i'(k+1) \leq J_i^*(k), \quad i \in \mathcal{V}. \quad (37)$$

Then, we consider the situation when the topology is changed, which means the neighbor set  $N_i(k) \neq N_i(k+1)$ . Define a virtual subproblem cost value at time  $k+1$  as  $J_i'^k(k+1)$ , which is characterized based on  $i \cup N_i(k)$  and the corresponding feasible solution (33) at  $k+1$ , the following inequation holds according to (33):

$$J_i'^k(k+1) \leq J_i^*(k), \quad i \in \mathcal{V}. \quad (38)$$

In addition, based on the feasible solution discussion in 1), we can compose a solution based on (33) under the new neighbor set  $N_i(k+1)$  and the corresponding cost value is denoted as  $J_i'^{k+1}(k+1)$  and a similar inequation can be obtained as

$$J_i^*(k+1) \leq J_i'^{k+1}(k+1), \quad i \in \mathcal{V}. \quad (39)$$

Based on (35), inequations (38) and (39) for all subproblems  $i \in \mathcal{V}$ , the following inequation with respect to the centralized objective function can be expressed in compact forms, respectively, as

$$J'^{k+1}(k) \leq J^{*k}(k) \quad (40)$$

$$J^*(k+1) \leq J'^{k+1}(k). \quad (41)$$

It can be concluded that despite the topology being changed, the centralized cost function is uniformly nonincreasing based on the control strategy described in Algorithm 1, that is,  $J^*(k+1) \leq J^*(k)$ ,  $k \in \mathbb{Z}^+$ . Based on the virtue of MPC controller as described in [39], the tracking error  $e(k)$  is also nonincreasing, which means that there exist  $\delta = e(0) > 0$  such that  $e(k) \leq \delta$  for all  $k > 0$ . The Lyapunov stability is proved.

## REFERENCES

- [1] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Syst.*, vol. 27, no. 2, pp. 71–82, Apr. 2007.
- [2] H. E. Li, C. Liu, Y. Zheng, J. Duan, and K. Li, "Distributed sensing, learning, and control for connected and automated vehicles," *Science*, pp. 41–43, 2017. [Online]. Available: <https://www.sciencemag.org/collections/reimagining-new-energy-vehicles-research-innovations-tongji-university>
- [3] J. Alonso-Mora, S. Baker, and D. Rus, "Multi-robot formation control and object transport in dynamic environments via constrained optimization," *Int. J. Robot. Res.*, vol. 36, no. 9, pp. 1000–1021, 2017.
- [4] Y. Liu and G. Nejat, "Multirobot cooperative learning for semiautonomous control in urban search and rescue applications," *J. Field Robot.*, vol. 33, no. 4, pp. 512–536, 2016.
- [5] H. M. La and W. Sheng, "Distributed sensor fusion for scalar field mapping using mobile sensor networks," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 766–778, Apr. 2013.
- [6] Y. Lyu, Q. Pan, C. Zhao, Y. Zhang, and J. Hu, "Vision-based UAV collision avoidance with 2D dynamic safety envelope," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 31, no. 7, pp. 16–26, Jul. 2016.
- [7] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *ACM SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 25–34, 1987.
- [8] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. Autom. Control*, vol. 51, no. 3, pp. 401–420, Mar. 2006.
- [9] H. Su, X. Wang, and Z. Lin, "Flocking of multi-agents with a virtual leader," *IEEE Trans. Autom. Control*, vol. 54, no. 2, pp. 293–307, Feb. 2009.
- [10] F. Cucker and J.-G. Dong, "A general collision-avoiding flocking framework," *IEEE Trans. Autom. Control*, vol. 56, no. 5, pp. 1124–1129, May 2011.

- [11] J. Zhu, J. Lu, and X. Yu, "Flocking of multi-agent non-holonomic systems with proximity graphs," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 1, pp. 199–210, Jan. 2013.
- [12] H. Fang, Y. Wei, J. Chen, and B. Xin, "Flocking of second-order multiagent systems with connectivity preservation based on algebraic connectivity estimation," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1067–1077, Apr. 2017.
- [13] Z. Bo and H. Duan, "Three-dimensional path planning for uninhabited combat aerial vehicle based on predator-prey pigeon-inspired optimization in dynamic environment," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 15, no. 99, pp. 356–371, Jan./Feb. 2017.
- [14] L. A. V. Reyes and H. G. Tanner, "Flocking, formation control, and path following for a group of mobile robots," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 4, pp. 1268–1282, Jul. 2015.
- [15] H. M. La, R. Lim, and W. Sheng, "Multirobot cooperative learning for predator avoidance," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 1, pp. 52–63, Jan. 2015.
- [16] S.-M. Hung and S. N. Givigi, "A Q-learning approach to flocking with UAVs in a stochastic environment," *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 186–197, Jan. 2017.
- [17] B. Alrifaae, K. Kostyszyn, and D. Abel, "Model predictive control for collision avoidance of networked vehicles using Lagrangian relaxation," *IFAC PapersOnLine*, vol. 49, no. 3, pp. 430–435, 2016.
- [18] Y. Zheng, S. E. Li, K. Li, F. Borrelli, and J. K. Hedrick, "Distributed model predictive control for heterogeneous vehicle platoons under unidirectional topologies," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 3, pp. 899–910, May 2017.
- [19] G. Ferrari-Trecate, L. Galbusera, M. P. E. Marciandi, and R. Scattolini, "Model predictive control schemes for consensus in multi-agent systems with single- and double-integrator dynamics," *IEEE Trans. Autom. Control*, vol. 54, no. 11, pp. 2560–2572, Nov. 2009.
- [20] X. Liu, Y. Shi, and D. Constantinescu, "Robust distributed model predictive control of constrained dynamically decoupled nonlinear systems: A contraction theory perspective," *Syst. Control Lett.*, vol. 105, pp. 84–91, Jul. 2017.
- [21] H.-T. Zhang, Z. Cheng, G. Chen, and C. Li, "Model predictive flocking control for second-order multi-agent systems with input constraints," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 6, pp. 1599–1606, Jun. 2015.
- [22] G.-P. Liu, "Predictive control of networked nonlinear multiagent systems with communication constraints," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.
- [23] B.-R. An, G.-P. Liu, and C. Tan, "Group consensus control for networked multi-agent systems with communication delays," *ISA Trans.*, vol. 76, pp. 78–87, May 2018.
- [24] S. E. Li *et al.*, "Dynamical modeling and distributed control of connected and automated vehicles: Challenges and opportunities," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 3, pp. 46–58, Jul. 2017.
- [25] F. Gao, D. F. Dang, S. S. Huang, and S. E. Li, "Decoupled robust control of vehicular platoon with identical controller and rigid information flow," *Int. J. Autom. Technol.*, vol. 18, no. 1, pp. 157–164, 2016.
- [26] P. Trodden and A. Richards, "Cooperative distributed MPC of linear systems with coupled constraints," *Automatica*, vol. 49, no. 2, pp. 479–487, 2013.
- [27] L. Zhou and S. Li, "Distributed model predictive control for multi-agent flocking via neighbor screening optimization," *Int. J. Robust Nonlinear Control*, vol. 27, no. 9, pp. 1690–1705, 2017.
- [28] P. Wang and B. Ding, "A synthesis approach of distributed model predictive control for homogeneous multi-agent system with collision avoidance," *Int. J. Control*, vol. 87, no. 1, pp. 52–63, 2014.
- [29] L. Dai, Q. Cao, Y. Xia, and Y. Gao, "Distributed MPC for formation of multi-agent systems with collision avoidance and obstacle avoidance," *J. Frankl. Inst.*, vol. 354, no. 4, pp. 2068–2085, 2017.
- [30] K. Huang and N. D. Sidiropoulos, "Consensus-ADMM for general quadratically constrained quadratic programming," *IEEE Trans. Signal Process.*, vol. 64, no. 20, pp. 5297–5310, Oct. 2016.
- [31] T. H. Summers and J. Lygeros, "Distributed model predictive consensus via the alternating direction method of multipliers," in *Proc. 50th Annu. Allerton Conf. Commun. Control Comput.*, 2012, pp. 79–84.
- [32] N. P. Bhatia and G. P. Szegö, *Stability Theory of Dynamical Systems*. Berlin, Germany: Springer-Verlag, 1970.
- [33] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *J. Mach. Learn. Res.*, vol. 17, no. 83, pp. 1–5, 2016.
- [34] J. Park and S. Boyd. (Mar. 2017). *General Heuristics for Nonconvex Quadratically Constrained Quadratic Programming*. [Online]. Available: arXiv:1703.07870
- [35] J. A. Preiss, W. Honig, G. S. Sukhatme, and N. Ayanian, "CrazySwarm: A large nano-quadcopter swarm," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 3299–3304.
- [36] *Vicon Motion Capture System*. Accessed: Jun. 8, 2017. [Online]. Available: <https://www.vicon.com>
- [37] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [38] A. Richards and J. P. How, "Model predictive control of vehicle maneuvers with guaranteed completion time and robust feasibility," in *Proc. Amer. Control Conf.*, vol. 5, Jun. 2003, pp. 4034–4040.
- [39] J. B. Rawlings, D. Bonne, J. B. Jorgensen, A. N. Venkat, and S. B. Jorgensen, "Unreachable setpoints in model predictive control," *IEEE Trans. Autom. Control*, vol. 53, no. 9, pp. 2209–2215, Oct. 2008.



**Yang Lyu** (S'18) received the B.Eng. degree from Guizhou University, Guiyang, China, in 2012, and the Ph.D. degree in control theory and applications from Northwestern Polytechnical University, Xi'an, China, in 2019.

He is a Postdoctoral Research Fellow with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His current research interests include control and estimation theory and its application to autonomous and intelligent systems.



**Jinwen Hu** (M'12) received the B.Eng. and M.Eng. degrees from Northwestern Polytechnical University (NPU), Xi'an, China, in 2005 and 2008, respectively, and the Ph.D. degree from Nanyang Technological University, Singapore.

He is currently an Associate Professor with the School of Automation, NPU. His current research interests include multiagent system, distributed control, unmanned vehicles, and information fusion.



**Ben M. Chen** (M'11) received the B.Sc. degree in mathematics and computer science from Xiamen University, Xiamen, China, in 1983, the M.Sc. degree in electrical engineering from Gonzaga University, Spokane, WA, USA, in 1988, and the Ph.D. degree in electrical and computer engineering from Washington State University, Pullman, WA, USA, in 1991.

He is currently a Professor with the Department of Mechanical and Automation Engineering, Chinese University of Hong Kong, Hong Kong. He was a

Provost's Chair Professor with the Department of Electrical and Computer Engineering, National University of Singapore (NUS), Singapore, where he was also serving as the Director of Control, Intelligent Systems and Robotics Area, and the Head of Control Science Group, Temasek Laboratories@NUS. His current research interests include unmanned systems, robust control, control applications, and financial market modeling.



**Chunhui Zhao** (F'07) received the B.E. degree from the School of Mechanics, Southwestern Jiaotong University, Chengdu, China, in 1996, and the M.E. and Ph.D. degrees from the School of Automation, Northwestern Polytechnical University (NPU), Xi'an, China, in 2003 and 2008, respectively.

He is currently an Associate Professor with the School of Automation, NPU. His current research interests include vision-based navigation, image processing, and target tracking.



**Quan Pan** (M'05) received the B.E. degree from the Huazhong Institute of Technology, Wuhan, China, in 1991, and the M.E. and Ph.D. degrees from the School of Automation, Northwestern Polytechnical University (NPU), Xi'an, China, in 1991 and 1997, respectively.

He is currently a Professor with the School of Automation, NPU. His current research interests include information fusion, multiscale estimation theory, target tracking, and image processing.