IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 9 October 2023, accepted 22 October 2023, date of publication 3 November 2023, date of current version 21 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3329232

RESEARCH ARTICLE

Distributed MPC for Trajectory Tracking and Formation Control of Multi-UAVs With Leader-Follower Structure

TIANLAI XU[1], JINLONG LIU[1], ZEXU ZHANG[1], GUODONG CHEN[1], DI CUI[2], AND HUIPING LI[2], (Senior Member, IEEE)

[1]School of Astronautics, Harbin Institute of Technology, Harbin 150080, China
[2]School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China

Corresponding author: Jinlong Liu (liujinlong_hit@163.com)

ABSTRACT Multiple Unmanned Aerial Vehicle (UAV) cooperation systems, such as flocking, consensus, formation control have a wide range of applications in monitoring, mapping, and target tracking. Optimal cooperative control of such systems is particularly important to increase their working efficiency. This paper studies the optimal trajectory tracking and formation control problems for multi-UAVs with a leader-follower structure, and the distributed model predictive control (MPC) scheme based formation control method is proposed. In particular, a novel MPC strategy is firstly designed for the leader modelled by the nonlinear Newton-Euler equations, to generate a feasible tracking trajectory for the formation systems. Then, by separating the system dynamics of the followers into the translation motion and the rotation motion, a two-layer distributed MPC formation control algorithm is designed to reduce the computation and communication loads, while only requiring limited information of the neighbors' states. Finally, simulation and comparison studies verify the effectiveness of the designed algorithms.

INDEX TERMS UAVs, model predictive control, leader-follower control.

I. INTRODUCTION

Research on behaviors of multiple unmanned aerial vehicle (UAV) cooperation systems has gained much attention from the communities in aeronautics and robotics, mainly due to its significantly improved flexibility and adaptability compared with the single agent [1], [2], [3]. Cooperative trajectory tracking of multi-UAVs, as one of the most controversial issues in cooperative control problems, is capable of simultaneously realizing trajectory tracking and formation control, showing great potential in practical applications such as border patrol, combat support, cargo transportation, and forest detection [4], [5], [6], [7], [8]. In this paper, to further improve the working efficiency, we investigate the optimal cooperative trajectory tracking problem of multi-UAVs.

In practice, the nonlinear, high coupling, and nonholonomic system dynamics of a UAV make it difficult to design the trajectory tracking controller. To address this issue, numerous control strategies for single UAV tracking have been developed. Existing results are classified into three types: linear control strategies, nonlinear control strategies, and intelligent control strategies. PID [9], LQR [10] and H_∞ [11] are typical linear control strategies, where the system model is linearized to reduce computation complexity. To overcome the performance degradation raised by the inaccuracy of the linear model, some nonlinear control strategies such as sliding mode control [12], backstepping control [13] and state feedback linearization approaches [14] are investigated. Because of the advantages in handling parameter adaptive adjustment and nonlinear functional approximation, intelligent control strategies including neural network [15], fuzzy logic system [16] and reinforcement learning [17] are also being studied. However, these control techniques either ignore the widely existed constraints (input saturation), or do not optimize the system performance, resulting in the control strategies being practically infeasible and consuming unnecessary resources.

Model predictive control (MPC) provides an inspiring solution to the optimal trajectory tracking of UAV. The main idea is to obtain an optimal control input sequence by

The associate editor coordinating the review of this manuscript and approving it for publication was Jie Gao.

solving an open-loop constrained optimization problem at each sampling instant. Then, only the first element of the control sequence is applied to the system. This mechanism endows MPC with the ability to control the system with high nonlinearity and negligible state/control input constraints, Motivated by these benefits, trajectory tracking of UAV using MPC has received a lot of attention [18], [19], [20], [21], [22]. In [20], an MPC approach was developed for a UAV to tracking a flat trajectory, with the system model linearized at several waypoints. Islam et al. designed an MPC tracking algorithm for a quaternion orientation based quadrotor, which effectively avoids the gimmal lock phenomenon in vertical take off and landing (VTOL) [21].

To realize the cooperative trajectory tracking of multi-UAVs, additional formation controller should also be considered. So far, a large amount of control techniques have been researched for the formation control of multi-UAVs, with typical schemes including the consensus-based method, the virtual structure method, the algebraic graph theory and the leader-follower strategy [23]. For example, time-varying formations were achieved by constructing a consensus-based controller in [24]. Zhou and Schwager proposed a formation controller using virtual structure method that enables a group of micro aerial vehicles to perform agile maneuvers in a fixed formation [25]. Huang et al. proposed a graph theory based formation control strategy to handle the formation control problem among heterogeneous multi-agents [26]. Based on the above techniques, the formation control methods in the famework of the distributed MPC are also investigated to endow the controller with optimized performance [27], [28], [29].

It should be noted that most works on multi-UAV formation control focus on the formation producing while ignoring the trajectory tracking of the overall system. Although the distributed MPC algorithms in [28] and [29] consider the optimal cooperative trajectory tracking problem of multi-UAVs, each agent is required to constantly perceive the information of the reference trajectory, placing high demands on the agent's hardware configuration. As a result, in this paper, we address the issue by employing the leader-follower based optimal control method, in which only the leader UAV is required to be sophisticated, allowing it to constantly track the reference and produce a feasible trajectory while taking the nonlinear system dynamics into consideration. Then, using a resource-aware distributed MPC formation controller, the multiple low-cost follower agents keep a desired distance with the leader UAV. The main contributions of this paper are as follows:

- To increase the accuracy of the UAV's state trajectory and cope with the control input constraints, a new MPC trajectory tracking algorithm is designed for the leader UAV with system dynamics directly regulated by the rotor's angular velocity. This differs from the traditional simplified system dynamics in [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], and [22], where the control inputs are taken as force and torques while neglecting the coupling among the input variables.
- For the follower agents, two-layer distributed MPC formation control methods with limited neighbor information are proposed to reduce communication load. We separate the system dynamics into the translational motion in the upper level and the rotation motion in the lower layer using the linear parameter varying (LPV) method. The communication burden is thereby decreased in two aspects. A distributed MPC approach with limited information about neighbor agents is designed to guarantee position formation in the upper level. A decentralized MPC without communication is designed to achieve the attitude stabilization in the lower layer.

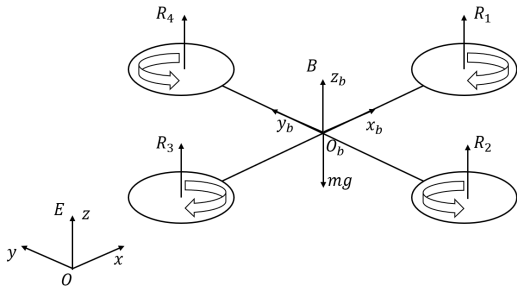

FIGURE 1. The basic configuration of quadrotor.

The remainder of this paper is organized as follows: Section II establishes the system dynamics and introduces the control object. In Section III, MPC based optimal cooperative trajectory tracking controller is designed. Then, simulation studies are provided in Section IV. Finally, Section V makes the conclusions.

II. PROBLEM FORMULATION

Consider the multiple UAV systems composed of a leader quadrotor and N homogeneous follower quadrotors. We assume that all the quadrotors have identical structures shown in Fig. 1, thus following the same modeling principle. For each quadrotor, the four propellers are located in a cross configuration and turn in opposite directions. Specifically, the front propeller 1 and the back propeller 3 turn in the clockwise direction, while the right propeller 2 and the left propeller 4 turn in the counterclockwise direction. By varying the angular velocity of the rotors, one can change the lift force and the torques applied on the quadrotor, hence activating quadrotor motion in 6 degrees of freedom.

In this paper, the leader-follower structure is employed to reduce the cooperative trajectory tracking cost, allowing a sophisticated leader quadrotor to guide a swarm of low-cost follower quadrotors. Note that the leader quadrotor is critical to the formation, since it constantly tracks the reference trajectory and provides guidance to the follower agents. Therefore, in this section, we develop the Newton-Euler equa-

tions as the leader's model, and we use a simplified model for the follower by considering the hardware configuration.

A. LEADER UAV MODEL

To facilitate model the system, two basic coordinates, named as earth-fixed frame $\boldsymbol{E}$ and body-fixed frame $\boldsymbol{B}$, are used, and the following assumption is made:

Assumption 1: (i) The quadrotor is a symmetrical rigid body; (ii) the mass and moment of inertia of the quadrotor do not change; (iii) the geometric center coincides with its center of gravity; (iv) the center of gravity coincides with the origin of the body-fixed frame.

Let $\boldsymbol{p} = [x, y, z]^T$ and $\boldsymbol{v} = [\dot{x}, \dot{y}, \dot{z}]^T$ be the position and the linear velocity of the quadrotor in the earth-fixed frame $\boldsymbol{E}$, respectively. Let $\boldsymbol{\theta} = [\phi, \theta, \psi]^T$ and $\boldsymbol{\omega} = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$ be the attitude and the angular velocity of the quadrotor in the body-fixed frame $\boldsymbol{B}$, respectively. The elements in $\boldsymbol{\theta}$ separately denote the roll angle, pitch angle and yaw angle along the axises $o_b x_b$, $o_b y_b$ and $o_b z_b$. Then, the kinematic model of the quadrotor is:

$$\begin{aligned} \dot{\boldsymbol{p}} &= \boldsymbol{v}, \\ \dot{\boldsymbol{\theta}} &= \boldsymbol{\omega}. \end{aligned} \tag{1}$$

Following the instruction of Newton-Eular equations, the dynamic model of the quadrotor can be formulated as [30]:

$$\dot{\boldsymbol{v}} = \begin{cases} \cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi)F/m, \\ \cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi)F/m, \\ (\cos\phi \cos\theta)F/m - g, \end{cases}$$

$$\dot{\boldsymbol{\omega}} = \begin{cases} \dot{\theta}\dot{\psi}(\frac{I_y - I_z}{I_x}) - \frac{J_r}{I_x}\dot{\theta}\Omega + \frac{1}{I_x}\tau_x, \\ \dot{\phi}\dot{\psi}(\frac{I_z - I_x}{I_y}) + \frac{J_r}{I_y}\dot{\phi}\Omega + \frac{1}{I_y}\tau_y, \\ \dot{\theta}\dot{\phi}(\frac{I_x - I_y}{I_z}) + \frac{1}{I_z}\tau_z, \end{cases} \tag{2}$$

where I_x, I_y and I_z are the components of the body inertial in the directions of the axis $o_b x_b$, $o_b y_b$ and $o_b z_b$, respectively; J_r indicates the rotor inertial; g is the gravitational acceleration; F and $\boldsymbol{\tau} = [\tau_x, \tau_y, \tau_z]^T$ represent the lift force and the torque applied on the quadrotor, respectively. Let $\Omega_i, i = 1, 2, 3, 4$ denote the rotor's angular velocity, we have

$$\Omega = \Omega_2 + \Omega_4 - \Omega_1 - \Omega_3 \tag{3}$$

and

$$\begin{aligned} F &= b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2), \\ \tau_x &= lb(\Omega_4^2 - \Omega_2^2), \\ \tau_y &= lb(\Omega_3^2 - \Omega_1^2), \\ \tau_z &= d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2), \end{aligned} \tag{4}$$

respectively. Here, l is the distance between the rotor's center and the geometric center of the quadrotor; b, d denote the thrust and drag factors.

In [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], and [22], the variables F, τ_x, τ_y and τ_z are taken as the four separate control inputs. However, it should be noticed that there is coupling among the input variables because of the term Ω in (3). To resolve this problem, we directly select the leader's control input $\boldsymbol{u}_l = [\Omega_1, \Omega_2, \Omega_3, \Omega_4]^T$ in this paper. System dynamics of the leader can be derived by substituting the equations in (3) and (4) into (2), improving control accuracy.

B. FOLLOWER UAV MODEL

Due to the limitation of the hardware configuration, we further divide system dynamics into the translation motion and the rotation motion for the followers to reduce the complexity of the model.

For the translation motion, define the control input $\boldsymbol{u}_{tr} = [u_{p_x}, u_{p_y}, u_{p_z}]^T$, where

$$\begin{aligned} u_{p_x} &= (\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi)F/m, \\ u_{p_y} &= (\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi)F/m, \\ u_{p_z} &= (\cos\phi \cos\theta)F/m - g. \end{aligned} \tag{5}$$

The system model in the upper level can be formulated by the following linear system:

$$\dot{\boldsymbol{x}}_{tr} = \begin{bmatrix} \boldsymbol{0}_3 & \boldsymbol{I}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 \end{bmatrix} \boldsymbol{x}_{tr} + \begin{bmatrix} \boldsymbol{0}_3 \\ \boldsymbol{I}_3 \end{bmatrix} \boldsymbol{u}_{tr}, \tag{6}$$

where $\boldsymbol{x}_{tr} = [\boldsymbol{p}^T, \boldsymbol{v}^T]^T$.

For the rotation motion, LPV method in [31] is adopted to decrease the modelling error. The detailed formulation of the attitude system is

$$\dot{\boldsymbol{x}}_{ro} = \begin{bmatrix} \boldsymbol{0}_3 & \boldsymbol{I}_3 \\ \boldsymbol{0}_3 & A_{22} \end{bmatrix} \boldsymbol{x}_{ro} + \begin{bmatrix} \boldsymbol{0}_3 \\ B_{21} \end{bmatrix} \boldsymbol{u}_{ro} \tag{7}$$

with

$$\begin{aligned} \boldsymbol{x}_{ro} &= [\boldsymbol{\theta}^T, \boldsymbol{\omega}^T]^T, \\ \boldsymbol{u}_{ro} &= [F, \tau_x, \tau_y, \tau_z, \Omega]^T, \\ A_{22} &= \begin{bmatrix} 0 & \frac{I_y - I_z}{2I_x}\dot{\psi} & \frac{I_y - I_z}{2I_x}\dot{\theta} \\ \frac{I_z - I_x}{2I_y}\dot{\psi} & 0 & \frac{I_z - I_x}{2I_y}\dot{\phi} \\ \frac{I_x - I_y}{2I_z}\dot{\theta} & \frac{I_x - I_y}{2I_z}\dot{\phi} & 0 \end{bmatrix}, \\ B_{21} &= \begin{bmatrix} 0 & \frac{1}{I_x} & 0 & 0 & -\frac{J_r}{I_x}\dot{\theta} \\ 0 & 0 & \frac{1}{I_y} & 0 & \frac{J_r}{I_y}\dot{\phi} \\ 0 & 0 & 0 & \frac{1}{I_z} & 0 \end{bmatrix}. \end{aligned} \tag{8}$$

Remark 1: Note that A_{22} and B_{21} are time-varying parameters. By linearizing the system at the operation points with a small time interval, (7) can well approximate the original nonlinear systems.

C. CONTROL OBJECT

For a sophisticated leader modelled by (1)-(4) and N low cost followers described by (6) and (7), we firstly aim at designing an optimal trajectory tracking controller for the leader that generates a feasible tracking trajectory while satisfying the control input constraints. Then, a two layer distributed leader-follower formation controller is designed to

realize that (i) the overall system forms a prescribed spatial configuration with optimized performance; (ii) only limited information of the neighbors' states is transmitted through the communication networks.

III. MODEL PREDICTIVE CONTROLLER DESIGN

In order to realize the control object while avoiding unnecessary resource consumption, MPC based optimal trajectory tracking controller for the leader is designed in Section III-A, and the distributed MPC formation controller for the follower is studied in Section III-B.

A. TRAJECTORY TRACKING

For the ease of computer processing, the discrete-time version of the nonlinear leader's model is firstly required. Then, in standard MPC, an constrained optimal control problem (OCP) minimizing the a cost function over a prediction horizon is recursively solved at each sampling instant.

Considering a sampling interval T, the discrete-time system state $\boldsymbol{x}_l(k)$ and control input of the leader at time step k are given by

$$\boldsymbol{x}_l(k) = \begin{bmatrix} x_l^1(k) \\ x_l^2(k) \\ \vdots \\ x_l^{12}(k) \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}_l(kT) \\ \boldsymbol{\theta}_l(kT) \\ \boldsymbol{v}_l(kT) \\ \boldsymbol{\omega}_l(kT) \end{bmatrix},$$

$$\boldsymbol{u}_l(k) = \begin{bmatrix} u_l^1(k) \\ u_l^2(k) \\ u_l^3(k) \\ u_l^4(k) \end{bmatrix} = \begin{bmatrix} \Omega_1(kT) \\ \Omega_2(kT) \\ \Omega_3(kT) \\ \Omega_4(kT) \end{bmatrix}.$$

Using Eular explicit method, we have

$$\begin{aligned} \boldsymbol{x}_l(k+1) &= f(\boldsymbol{x}_l(k), \boldsymbol{u}_l(k)) \\ \boldsymbol{y}_l(k) &= \boldsymbol{C}_l \boldsymbol{x}_l(k) = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{I}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 \end{bmatrix} \boldsymbol{x}_l(k), \end{aligned} \tag{9}$$

where

$$f(\boldsymbol{x}_l(k), \boldsymbol{u}_l(k)) = \begin{bmatrix} x_l^1(k) + Tx_l^7(k) \\ x_l^2(k) + Tx_l^8(k) \\ x_l^3(k) + Tx_l^9(k) \\ x_l^4(k) + Tx_l^{10}(k) \\ x_l^5(k) + Tx_l^{11}(k) \\ x_l^6(k) + Tx_l^{12}(k) \\ x_l^7(k) + T[\cos(x_l^4(k))\sin(x_l^5(k))\cos(x_l^6(k)) \\ \quad + \sin(x_l^4(k))\sin(x_l^6(k))]F(k)/m \\ x_l^8(k) + T[\cos(x_l^4(k))\sin(x_l^5(k))\sin(x_l^6(k)) \\ \quad - \sin(x_l^4(k))\cos(x_l^6(k))]F(k)/m \\ x_l^9(k + T[(\cos(x_l^4(k))\cos(x_l^5(k)))F(k)/m - g)] \\ x_l^{10}(k) + T[x_l^{11}(k)x_l^{12}(k)(\frac{I_y - I_z}{I_x}) - \frac{J_r}{I_x}x_l^{11}(k)\Omega(k) + \frac{\tau_x(k)}{I_x}] \\ x_l^{11}(k) + T[x_l^{10}(k)x_l^{12}(k)(\frac{I_z - I_x}{I_y}) + \frac{J_r}{I_y}x_l^{10}(k)\Omega(k) + \frac{\tau_y(k)}{I_y}] \\ x_l^{12}(k) + T[x_l^{10}(k)x_l^{11}(k)(\frac{I_x - I_y}{I_z}) + \frac{1}{I_z}\tau_z(k)] \end{bmatrix} \tag{10}$$

$\boldsymbol{y}_l(k)$ is the output of the leader at time step k, and

$$\begin{cases} F(k) = b(u_l^1{}^2(k) + u_l^2{}^2(k) + u_l^3{}^2(k) + u_l^4{}^2(k)), \\ \tau_x(k) = lb(u_l^4{}^2(k) - u_l^2{}^2(k)), \\ \tau_y(k) = lb(u_l^3{}^2(k) - u_l^1{}^2(k)), \\ \tau_z(k) = d(u_l^2{}^2(k) + u_l^4{}^2(k) - u_l^1{}^2(k) - u_l^3{}^2(k)). \end{cases} \tag{11}$$

With the time-varying reference trajectory $\boldsymbol{y}_r(k) = [\boldsymbol{p}_r(k), \boldsymbol{\theta}_r(k)]^T$ and the prediction model in (9), the cost function measuring the tracking cost and control energy is designed as follows:

$$\begin{aligned} &J(\boldsymbol{u}_l(\cdot;k); \boldsymbol{x}_l(k), \boldsymbol{y}_r(k)) \\ &\triangleq \sum_{i=0}^{N-1} [\left\|\boldsymbol{y}_l(k+i;k) - \boldsymbol{y}_r(k+i)\right\|_{Q_l}^2 + \left\|\boldsymbol{u}_l(k+i;k) - \boldsymbol{u}_e\right\|_{R_l}^2] \\ &\quad + \left\|\boldsymbol{y}_l(k+N;k) - \boldsymbol{y}_r(k+N)\right\|_{P_l}^2, \end{aligned} \tag{12}$$

where $\boldsymbol{u}_l(k+i;k), i = 0, \cdots, N-1$ is the feasible control input at time step $k+i$ predicted at k, and $\boldsymbol{y}_l(k+i;k), i = 0, \cdots, N$ denotes the corresponding predicted output starting from the initial value $\boldsymbol{y}_l(k) = \boldsymbol{C}_l \boldsymbol{x}_l(k)$; $\boldsymbol{u}_e$ is the control input when the leader stays at a static hovering state, which guarantees that the leader will adjust its state near the equilibrium. N is the prediction horizon. The weighting matrices Q_l, R_l and P_l are all positive definite.

Using (12), the OCP is given by

Problem $\mathcal{P}1$:

$$\begin{aligned} \boldsymbol{U}_l^*(k) = \arg \min_{\boldsymbol{u}_l(\cdot;k)} & J(\boldsymbol{u}_l(\cdot;k); \boldsymbol{x}_l(k), \boldsymbol{y}_r(k)) \\ s.t.\ & \boldsymbol{x}_l(k+i+1;k) = f(\boldsymbol{x}_l(k+i;k), \boldsymbol{u}_l(k+i;k)) \\ & \boldsymbol{x}_l(k;k) = \boldsymbol{x}_l(k) \\ & \boldsymbol{y}_l(k+i;k) = \boldsymbol{C}_l \boldsymbol{x}_l(k+i;k) \\ & \boldsymbol{x}_l(k+i;k) \in \mathcal{X}_l \\ & \boldsymbol{u}_l(k+i;k) \in \mathcal{U}_l. \end{aligned}$$

Here, $\boldsymbol{U}_l^*(k) = [\boldsymbol{u}_l^*(k;k), \boldsymbol{u}_l^*(k+1;k), \cdots, \boldsymbol{u}_l^*(k+N-1;k)]^T$ is the optimal control input sequence predicted at time step k; $\boldsymbol{u}_l(k+i;k) \in \mathcal{X}_l$ and $\mathcal{U}_l$ denote the state constraint set and the input constraint set, respectively.

After solving Problem $\mathcal{P}1$, the current control input of the leader is determined by the first element of $\boldsymbol{U}_l^*(k)$, that is

$$\boldsymbol{u}_l^*(k) = \boldsymbol{u}_l^*(k;k) \tag{13}$$

The block diagram of the model predictive trajectory tracking controller for the leader is detailed in Fig. 2.

Remark 2: Since the discretization of the nonlinear system dynamics will lead to the model inaccuracy, adopting such a model may cause system instability especially when the sampling interval T is large. One way to avoid this issue is to adopt the high order accuracy difference schemes such as the Runge-Kutta method to reduce the error of discretization. A different approach would be to incorporate an event triggered mechanism as in [32], where the control law is updated once the discrepancy between the actual and

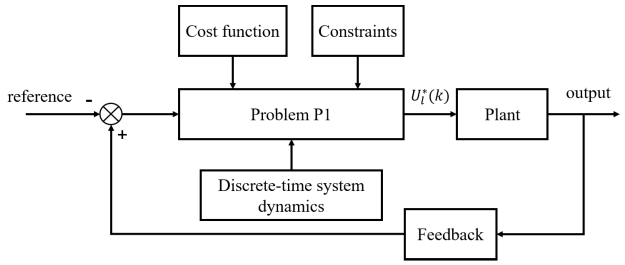

FIGURE 2. The block diagram of the model predictive trajectory tracking controller.

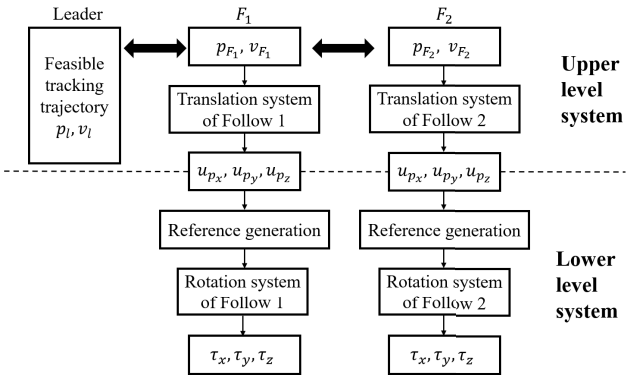

FIGURE 3. Two-layer formation control structure.

desired system performance reaches a threshold. The state estimator designed in [33] also provides a inspiring solution to suppress the influence of model inaccuracy.

Remark 3: Note that in comparison with conventional methods, the advantage of MPC strategy lies two points. First, it can do online optimization and achieve optimal control performance. Second, it can add the system constraints directly into the optimization problem which can easily fulfill system constraints. The disadvantage is that MPC requires more computational resources.

B. FORMATION CONTROL

In this subsection, two-layer distributed MPC formation controller for the low-cost follower with limited neighbor information is designed. We first show the two-layer structure of the proposed control strategy, then present the distributed MPC strategy for the position formation in the upper level and design the decentralized MPC attitude stabilization method in the low level.

(Two-layer control structure) For the ease of presentation, our two-layer control structure is shown in Fig. 4, where the multi-quadrotors including one leader and two followers. The cooperative behaviors are enforced for the translation system of the followers. In the upper level, only the leader's position and linear velocity are required. A reference generating operator then transforms the obtained control input into the reference attitude. Finally, using a decentralized MPC strategy, the rotation system in the lower level will track the reference command.

(Upper level controller design) In the upper level, the discrete-time system dynamics of the translation motion is first established for the follower i.

Similar to the method in Section III-A, the discrete-time system state and the control input are given by

$$\boldsymbol{x}_{tr}^i(k) = \begin{bmatrix} x_{tr}^{1i}(k) \\ \vdots \\ x_{tr}^{6i}(k) \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}^i(kT) \\ \boldsymbol{v}^i(kT) \end{bmatrix},$$

$$\boldsymbol{u}_{tr}^i(k) = \begin{bmatrix} u_{tr}^{1i}(k) \\ u_{tr}^{2i}(k) \\ u_{tr}^{3i}(k) \end{bmatrix} = \begin{bmatrix} u_{p_x}^i(kT) \\ u_{p_y}^i(kT) \\ u_{p_z}^i(kT) \end{bmatrix}.$$

Using the Eular explicit method with the sampling internal T, we have

$$\begin{aligned} \boldsymbol{x}_{tr}^i(k+1) &= g(\boldsymbol{x}_{tr}^i(k), \boldsymbol{u}_{tr}^i(k)) \\ &= \begin{bmatrix} \boldsymbol{I}_3 & T\boldsymbol{I}_3 \\ \boldsymbol{0}_3 & \boldsymbol{I}_3 \end{bmatrix} \boldsymbol{x}_{tr}^i(k) + \begin{bmatrix} \boldsymbol{0}_3 \\ T\boldsymbol{I}_3 \end{bmatrix} \boldsymbol{u}_{tr}^i(k), \\ \boldsymbol{y}_{tr}^i(k) &= \boldsymbol{C}_{tr}\boldsymbol{x}_{tr}^i(k) = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_3 \end{bmatrix} \boldsymbol{x}_{tr}^i(k), \end{aligned} \tag{14}$$

where $\boldsymbol{y}_{tr}^i(k)$ is the output of the follower in the upper level at time step k. Then, the following cost function is defined.

$$\begin{aligned} &J(\boldsymbol{u}_{tr}^i(\cdot;k); \boldsymbol{x}_{tr}^i(k), \boldsymbol{y}_{tr}^j(\cdot;k)) \\ &\triangleq \sum_{j\in\mathcal{N}_i} \left\| \boldsymbol{y}_{tr}^i(k+N;k) - \boldsymbol{y}_{tr}^j(k+N;k) + d_{ij} \right\|_{P_{tr}^{ij}}^2 \\ &\quad + \sum_{h=0}^{N-1} [\sum_{j\in\mathcal{N}_i} \left\| \boldsymbol{y}_{tr}^i(k+h;k) - \boldsymbol{y}_{tr}^j(k+h;k) + d_{ij} \right\|_{Q_{tr}^{ij}}^2 \\ &\quad + \left\| \boldsymbol{u}_{tr}^i(k+h;k) \right\|_{R_{tr}^i}^2], \end{aligned} \tag{15}$$

where $\boldsymbol{u}_{tr}^i(k+h;k), h = 0, \cdots, N-1$ is the feasible control input at time step $k+h$ predicted at k for the follower i, and $\boldsymbol{y}_{tr}^i(k+h;k), h = 0, \cdots, N$ denotes the corresponding predicted output starting from the initial value $\boldsymbol{y}_{tr}^i(k) = \boldsymbol{C}_{tr}\boldsymbol{x}_{tr}^i(k)$; $\mathcal{N}_i$ is the neighbor set of the follower i. $\boldsymbol{y}_{tr}^j(\cdot;k)$ is the output sequence of neighbor j predicted at current time step k. However, the fact that each agent solves the optimization problem synchronously prevents agent i from accessing the neighbor information. Therefore, with the received information of neighbor j at the most recent time step through the communication network, i.e., $\boldsymbol{x}_{tr}^j(k-1)$ and $\boldsymbol{U}_{tr}^{j*}(k-1)$, the following assumed output sequence of neighbor j is used as the substitutions:

$$\hat{\boldsymbol{y}}_{tr}^j(k+h;k) = \boldsymbol{C}_{tr}\hat{\boldsymbol{x}}_{tr}^j(k+h;k), \tag{16}$$

where the assume state sequence $\hat{\boldsymbol{x}}_{tr}^j(k+h;k)$ evolves according to the initial state $\hat{\boldsymbol{x}}_{tr}^j(k;k) = g(\boldsymbol{x}_{tr}^j(k-1), \boldsymbol{u}_{tr}^{i*}(k-1;k-1))$ and the assumed predictive control input of j as follows.

$$\hat{\boldsymbol{u}}_{tr}^j(k+h;k) = \begin{cases} \boldsymbol{u}_{tr}^{j*}(k+h;k-1) & h = 1, \cdots, N-2 \\ \boldsymbol{u}_{tr}^{j*}(k+N-2;k-1) & h = N-1. \end{cases} \tag{17}$$

With the assumed output sequence of neighbor j in (16), an OCP is formulated for the follower agent i:

Problem $\mathcal{P}2$:

$$\begin{aligned}
&\boldsymbol{U}_{tr}^{i*}(k) = \arg\min_{\boldsymbol{u}_{tr}^{i}(\cdot;k)} J(\boldsymbol{u}_{tr}^{i}(\cdot;k); \boldsymbol{x}_{tr}^{i}(k), \hat{\boldsymbol{y}}_{tr}^{j}(\cdot;k))\\
&s.t.\ \boldsymbol{x}_{tr}^{i}(k+h+1;k) = g(\boldsymbol{x}_{tr}^{i}(k+h;k), \boldsymbol{u}_{tr}^{i}(k+h;k))\\
&\boldsymbol{x}_{tr}^{i}(k;k) = \boldsymbol{x}_{tr}^{i}(k)\\
&\boldsymbol{y}_{tr}^{i}(k+h;k) = \boldsymbol{C}_{tr}\boldsymbol{x}_{tr}^{i}(k+h;k)\\
&\boldsymbol{x}_{tr}^{i}(k+h;k) \in \mathcal{X}_{tr}\\
&\boldsymbol{u}_{tr}^{i}(k+h;k) \in \mathcal{U}_{tr}
\end{aligned} \tag{18}$$

where $\boldsymbol{U}_{tr}^{i*}(k) = [\boldsymbol{u}_{tr}^{*}(k;k), \boldsymbol{u}_{tr}^{*}(k+1;k), \cdots, \boldsymbol{u}_{tr}^{*}(k+N-1;k)]^T$ is the obtained optimal control input sequence. By solving Problem $\mathcal{P}2$, the first element of $\boldsymbol{U}_{tr}^{i*}(k)$ is applied to the agent i:

$$\boldsymbol{u}_{tr}^{i*}(k) = \boldsymbol{u}_{tr}^{*}(k;k). \tag{19}$$

It should be noted that transmitting neighbors' information $\boldsymbol{x}_{tr}^{j}(k-1)$ and $\boldsymbol{U}_{tr}^{j*}(k-1)$ puts a high requirement on the bandwidth of the communication network. To reduce the communication load, we design a new OCP, where the assumed output sequence of neighbor j is constructed under limited neighbors' information, that is, only the elements $\boldsymbol{x}_{tr}^{j}(k-1)$ and $\boldsymbol{u}_{tr}^{j*}(k-1;k-1)$ are required.

Problem $\mathcal{P}3$:

$$\begin{aligned}
&\boldsymbol{U}_{tr}^{i*}(k) = \arg\min_{\boldsymbol{u}_{tr}^{i}(\cdot;k)} J(\boldsymbol{u}_{tr}^{i}(\cdot;k); \boldsymbol{x}_{tr}^{i}(k), \hat{\boldsymbol{y}}_{tr}^{j}(\cdot;k))\\
&s.t.\ \boldsymbol{x}_{tr}^{i}(k+h+1;k) = g(\boldsymbol{x}_{tr}^{i}(k+h;k), \boldsymbol{u}_{tr}^{i}(k+h;k))\\
&\boldsymbol{x}_{tr}^{i}(k;k) = \boldsymbol{x}_{tr}^{i}(k)\\
&\boldsymbol{y}_{tr}^{i}(k+h;k) = \boldsymbol{C}_{tr}\boldsymbol{x}_{tr}^{i}(k+h;k)\\
&\boldsymbol{x}_{tr}^{i}(k+h;k) \in \mathcal{X}_{tr}\\
&\boldsymbol{u}_{tr}^{i}(k+h;k) \in \mathcal{U}_{tr}\\
&\hat{\boldsymbol{y}}_{tr}^{j}(k+h;k) = \boldsymbol{C}_{tr}\hat{\boldsymbol{x}}_{tr}^{j}(k+h;k)\\
&\hat{\boldsymbol{x}}_{tr}^{j}(k;k) = g(\boldsymbol{x}_{tr}^{j}(k-1), \boldsymbol{u}_{tr}^{j*}(k-1;k-1))\\
&\hat{\boldsymbol{x}}_{tr}^{j}(k+h+1;k) = g(\hat{\boldsymbol{x}}_{tr}^{j}(k+h;k), \boldsymbol{0}_3)
\end{aligned} \tag{20}$$

Here, we approximate $\hat{\boldsymbol{y}}_{tr}^{j}(k+h;k)$ by assuming that the state of neighbor agent j evolves in a constant control input.

(Lower level controller design) According to the definition of $\boldsymbol{u}_{tr}$ in (5), with the help of the counter transformation method, the optimal control input $\boldsymbol{u}_{tr}^{i*}$ obtained in the upper level can be transferred into the lift force and the attitude values as follows.

$$\begin{cases}
F_r^i(k) = m\sqrt{u_{tr}^{1i*2}(k) + u_{tr}^{2i*2}(k) + (u_{tr}^{3i*}(k)+g)^2}\\
\phi_r^i(k) = \arcsin(\dfrac{m}{F^r(k)}(u_{tr}^{1i*}(k)\sin\psi_r(k)\\
\qquad -u_{tr}^{2i*}(k)\cos\psi_r(k)))\\
\theta_r^i(k) = \arctan(\dfrac{1}{u_{tr}^{3i*}(k)+g}(u_{tr}^{1i*}(k)\cos\psi_r(k)\\
\qquad +u_{tr}^{2i*}(k)\sin\psi_r(k))).
\end{cases} \tag{21}$$

where ψ_r is a predetermined yaw angle. Then, we take the vector $\boldsymbol{\theta}_r^i(k) = [\phi_r^i(k), \theta_r^i(k), \psi_r(k)]^T$ as the reference attitude, which will be tracked by the rotation system in the lower level.

Different from the sampling interval T in the upper level, here, we further divide T into N_{inner} subintervals with the new sampling interval $\bar{T} = \frac{T}{N_{inner}}$. Then, the discrete-time state $\boldsymbol{x}_{ro}^{i}(\bar{k})$ and the control input are

$$\boldsymbol{x}_{ro}^{i}(\bar{k}) = \begin{bmatrix} x_{ro}^{1i}(\bar{k})\\ \vdots\\ x_{ro}^{6i}(\bar{k}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\theta}^{i}(\bar{k}\bar{T})\\ \boldsymbol{\omega}^{i}(\bar{k}\bar{T}) \end{bmatrix},$$

$$\boldsymbol{u}_{ro}^{i}(\bar{k}) = \begin{bmatrix} u_{ro}^{1i}(\bar{k})\\ u_{ro}^{2i}(\bar{k})\\ u_{ro}^{3i}(\bar{k})\\ u_{ro}^{4i}(\bar{k}) \end{bmatrix} = \begin{bmatrix} \tau_x^{i}(\bar{k}\bar{T})\\ \tau_y^{i}(\bar{k}\bar{T})\\ \tau_z^{i}(\bar{k}\bar{T})\\ \Omega^{i}(\bar{k}\bar{T}) \end{bmatrix}.$$

By applying Eular explicit method with the sampling internal $\bar{T}$ to the continuous-time system dynamics in (7), the discrete-time system dynamics are given by

$$\begin{aligned}
\boldsymbol{x}_{ro}^{i}(\bar{k}+1) &= h(\boldsymbol{x}_{ro}^{i}(\bar{k}), \boldsymbol{u}_{ro}^{i}(\bar{k}))\\
&= \begin{bmatrix} \boldsymbol{I}_3 & \bar{T}\boldsymbol{I}_3\\ \boldsymbol{0}_3 & \bar{A}_{22} \end{bmatrix}\boldsymbol{x}_{ro}^{i}(\bar{k}) + \begin{bmatrix} \boldsymbol{0}_3\\ \bar{T}B_{21} \end{bmatrix}\boldsymbol{u}_{ro}^{i}(\bar{k})\\
\boldsymbol{y}_{ro}^{i}(\bar{k}) &= \boldsymbol{C}_{ro}\boldsymbol{x}_{ro}^{i}(\bar{k}) = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_3 \end{bmatrix}\boldsymbol{x}_{ro}^{i}(\bar{k}),
\end{aligned} \tag{22}$$

with

$$\bar{A}_{22} = \begin{bmatrix} 1 & \frac{I_y-I_z}{2I_x}\bar{T}x_{ro}^{6i}(\bar{k}) & \frac{I_y-I_z}{2I_x}\bar{T}x_{ro}^{5i}(\bar{k})\\ \frac{I_z-I_x}{2I_y}\bar{T}x_{ro}^{6i}(\bar{k}) & 1 & \frac{I_z-I_x}{2I_y}\bar{T}x_{ro}^{4i}(\bar{k})\\ \frac{I_x-I_y}{2I_z}\bar{T}x_{ro}^{5i}(\bar{k}) & \frac{I_x-I_y}{2I_z}\bar{T}x_{ro}^{4i}(\bar{k}) & 1 \end{bmatrix}, \tag{23}$$

where $\boldsymbol{y}_{ro}^{i}(\bar{k})$ is the output of the follower in the lower lever at time step $\bar{k}$.

For the system dynamics in (22), define a cost function as

$$\begin{aligned}
J(\boldsymbol{u}_{ro}^{i}(\cdot;\bar{k}); \boldsymbol{x}_{ro}^{i}(\bar{k}), \boldsymbol{\theta}_r^{i}(k)) \triangleq &\left\|\boldsymbol{y}_{ro}^{i}(\bar{k}+\bar{N};\bar{k}) - \boldsymbol{\theta}_r^{i}(k)\right\|_{P_{ro}}^2 +\\
\sum_{h=0}^{\bar{N}-1} &\left\|\boldsymbol{y}_{ro}^{i}(\bar{k}+h;\bar{k}) - \boldsymbol{\theta}_r^{i}(k)\right\|_{Q_{ro}}^2 + \left\|\boldsymbol{u}_{ro}^{i}(\bar{k}+h;\bar{k}) - \boldsymbol{u}_r^{i}\right\|_{R_{ro}}^2,
\end{aligned} \tag{24}$$

where $\boldsymbol{u}_{ro}^{i}(\bar{k}+h;\bar{k}), h = 0, \cdots, \bar{N}-1$ is the future feasible control input predicted at time step $\bar{k}$ for the follower i, $\boldsymbol{u}_r^{i} = [F_r^{i}, 0, 0, 0, 0]^T$, $\boldsymbol{y}_{ro}^{i}(\bar{k}+h;\bar{k}), h = 0, \cdots, \bar{N}$ denotes the corresponding future output starting from the initial output $\boldsymbol{y}_{ro}^{i}(\bar{k}) = \boldsymbol{C}_{ro}\boldsymbol{x}_{ro}^{i}(\bar{k})$.

Next, the OCP for the lower level is presented as follows:

Problem $\mathcal{P}4$:

$$\begin{aligned}
&\boldsymbol{U}_{ro}^{i*}(\bar{k}) = \arg\min_{\boldsymbol{u}_{ro}^{i}(\cdot;\bar{k})} J(\boldsymbol{u}_{ro}^{i}(\cdot;\bar{k}); \boldsymbol{x}_{ro}^{i}(\bar{k}), \boldsymbol{\theta}_r^{i}(k))\\
&\quad s.t.\ \boldsymbol{x}_{ro}^{i}(\bar{k}+h+1;\bar{k}) = h(\boldsymbol{x}_{ro}^{i}(\bar{k}+h;\bar{k}), \boldsymbol{u}_{ro}^{i}(\bar{k}+h;\bar{k}))\\
&\quad \boldsymbol{x}_{ro}^{i}(\bar{k};\bar{k}) = \boldsymbol{x}_{ro}^{i}(\bar{k})\\
&\quad \boldsymbol{y}_{ro}^{i}(\bar{k}+h;\bar{k}) = \boldsymbol{C}_{ro}\boldsymbol{x}_{ro}^{i}(\bar{k}+h;\bar{k})\\
&\quad \boldsymbol{u}_{ro}^{i}(\bar{k}+h;\bar{k}) \in \mathcal{U}_{ro},
\end{aligned}$$

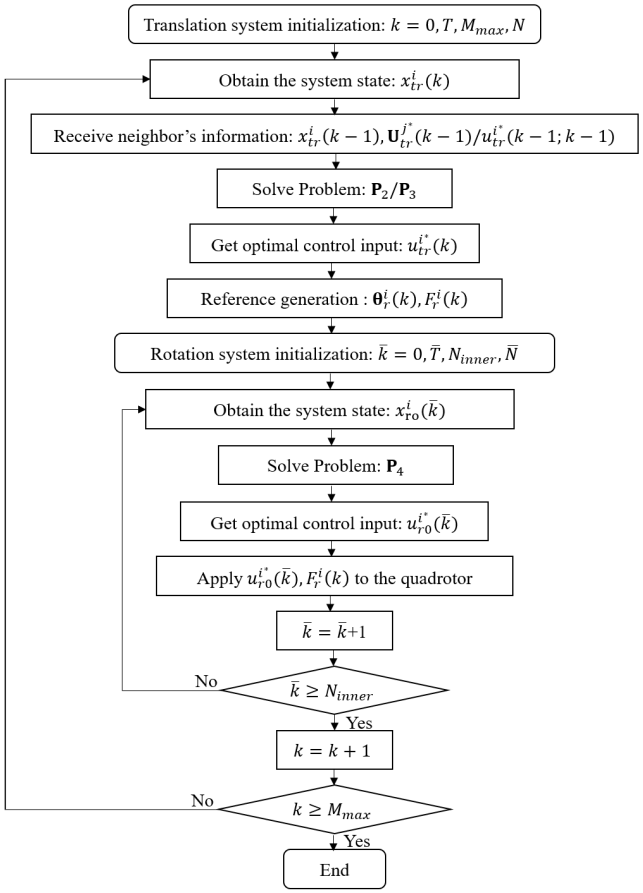

FIGURE 4. The flow chart of the two-layer distributed MPC formation algorithm.

where $\boldsymbol{U}_{ro}^{i*}(\bar{k}) = [\boldsymbol{u}_{ro}^{i*}(\bar{k};\bar{k}), \boldsymbol{u}_{ro}^{i*}(\bar{k}+1;\bar{k}), \cdots, \boldsymbol{u}_{ro}^{i*}(\bar{k}+\bar{N}-1;\bar{k})]^T$ is the optimal control input sequence. Then, the control input at time step $\bar{k}$ is determined by the first element of $\boldsymbol{U}_{ro}^{i*}(\bar{k})$:

$$\boldsymbol{u}_{ro}^{i*}(\bar{k}) = \boldsymbol{u}_{ro}^{i*}(\bar{k};\bar{k}). \tag{25}$$

Note that there do not exist any information of the neighbor agents. Therefore, the controller in the lower level works in a decentralized manner.

For the ease of practical implementation, the designed two-layer distributed MPC formation algorithm is detailed in Fig. 4, where M_{max} represents the maximum iteration numbers.

Remark 4: In comparison with one-layer method, the two-layer distributed MPC formation control methods have two advantages. Firstly, it can greatly reduce the computational burden due to separating a large optimization problem into small optimization problems. Secondly, it is more flexible to design, test and error diagnosis.

IV. SIMULATION

In this section, simulation examples are provided to show the effectiveness of the designed controller. Here, the multi-quadrotors system consists of one leader and two followers. The communication network among the agents is shown in Fig. 5, where $\mathcal{N}_1 = \{2, 3\}$, $\mathcal{N}_2 = \{1\}$ and $N_3 = \{1\}$. Here, we assume that the three quadrotors share the same system parameters listed in Table 1. Each rotor's angular velocity subject to the following constraints

$$0 \leqslant \Omega_i(k) \leqslant 600,\ i = 1, 2, 3, 4. \tag{26}$$

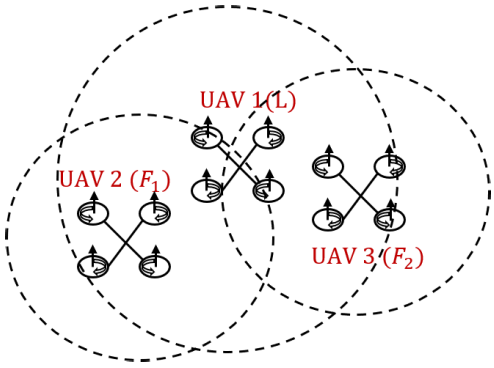

FIGURE 5. Communication network among the agents.

TABLE 1. Basic parameters for the quadrotor.

Symbol	value	Symbol	value
b	$1.105e^{-5}$	l	$0.225m$
d	$3.558e^{-7}$	m	$1.4kg$
I_x	$0.0211kg.m^2$	I_y	$0.0219kg.m^2$
I_z	$0.0366kg.m^2$	J_r	$1.287e^{-4}kg.m^2$

A. EXAMPLE ONE

We consider the formation control for the multi-quadrotors with the leader tracking a line formulated by

$$\begin{aligned} x_r(t) &= 0.2t, \\ y_r(t) &= 0.2t, \\ z_r(t) &= 0.2t, \\ \phi_r(t) &= 0, \theta_r(t) = 0, \psi_r(t) = 0. \end{aligned} \tag{27}$$

For the leader, the initial system state $\boldsymbol{x}_l(0) = [2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]^T$ and the sampling period $T = 0.3s$. $\boldsymbol{u}_e$ is calculated to be $[557.142, 557.142, 557.142, 557.142]^T$. The desired distances $d_{21} = [2, 0, 2]^T$ and $d_{31} = [-2, 0, 2]^T$. The prediction horizon N in the controller is set as 10. It should be noted that in MPC, a longer prediction horizon improves control performance by bringing more information to the optimization problem. A longer prediction horizon, on the other hand, increases the number of optimization variables, imposing even more stringent constraints on hardware setup. Additionally, the weighting matrices also influence the control performance. There is a trade-off between the output weighting matrices Q_l, P_l and the input weighting matrix R_l. Specifically, larger Q_l and P_l will make the system reach the reference trajectory faster, at the cost of consuming more control cost. Here, Q_l, P_l and

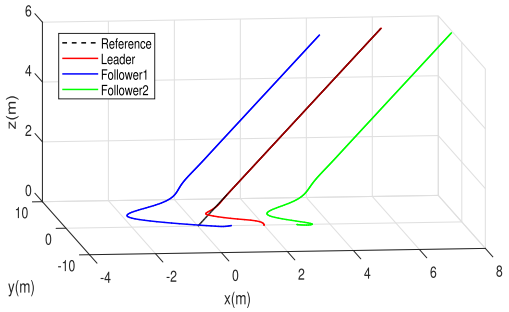

FIGURE 6. Trajectory of the three quadrotors under limited information (line).

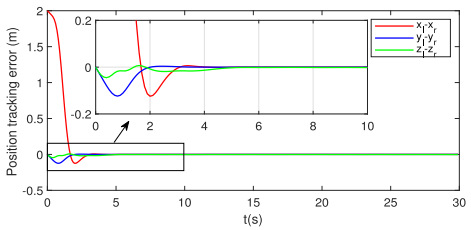

FIGURE 7. Trajectory tracking error of the leader with nonlinear model (line).

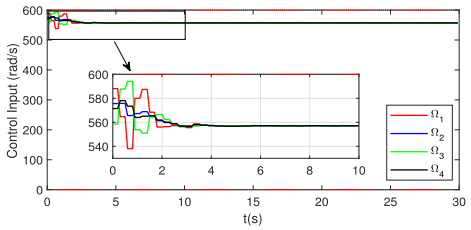

FIGURE 8. The control input of the leader with nonlinear model (line).

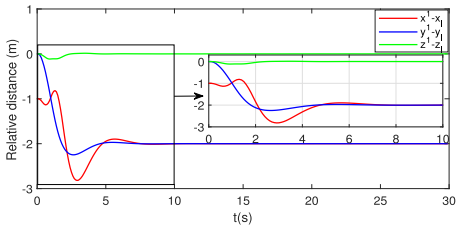

FIGURE 9. Relative distance between the follower 1 (two-layer model) and the leader (nonlinear model) under limited information (line).

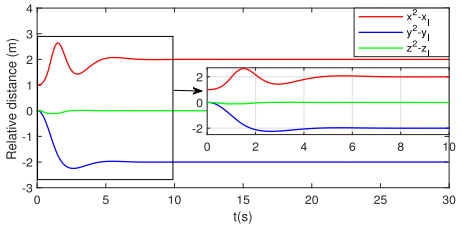

FIGURE 10. Relative distance between the follower 2 (two-layer model) and the leader (nonlinear model) under limited information (line).

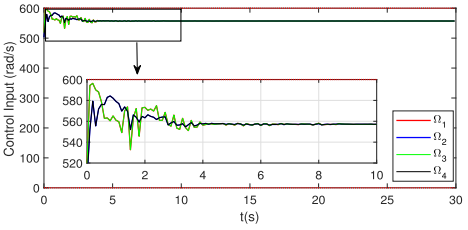

FIGURE 11. The control input of the follower 1 with two-layer model (line).

R_l are $\boldsymbol{I}_6$, $\boldsymbol{I}_6$ and $10^{-5}\boldsymbol{I}_6$, respectively; The system state is constrained by $\mathcal{X}_l = \{\boldsymbol{x}_l | [0,0,1]\cdot \boldsymbol{p}_l \geqslant 0\}$. For the followers, in the upper level, the initial states of the translation system are $\boldsymbol{x}_{tr}^2(0) = [1,0,0,0,0,0]^T$ and $\boldsymbol{x}_{tr}^3(0) = [3,0,0,0,0,0]^T$. The sampling period $T = 0.3s$; The weighting matrices in the cost function are set to be $Q_{tr}^{21} = Q_{tr}^{31} = \boldsymbol{I}_3$, $P_{tr}^{21} = P_{tr}^{31} = \boldsymbol{I}_3$ and $R_{tr}^2 = R_{tr}^3 = 10^{-1}\boldsymbol{I}_3$; The system state is constrained by $\mathcal{X}_{tr}^i = \{\boldsymbol{x}_{tr}^i | [0,0,1]\cdot \boldsymbol{p}^i \geqslant 0\}$, $i = 1.2$. In the lower level, the initial states of the rotation systems are $\boldsymbol{x}_{ro}^2(0) = I_5$ and $\boldsymbol{x}_{ro}^3(0) = [0,0,0,0,0,0]^T$. The sampling period $\bar{T} = 0.1s$, and the iteration numbers $N_{inner} = 3$. The prediction horizon $\bar{N}$ is given by 3. For the weighting matrices, $Q_{ro}^2 = Q_{ro}^3 = \boldsymbol{I}_3$, $P_{ro}^2 = P_{ro}^3 = \boldsymbol{I}_3$ and $R_{ro}^2 = R_{ro}^3 = I_5$.

We operate the simulation by MATLAB and solve OCP by *fmincon* function. In section III-B, two sets of OCPs have been developed, driving the agents to form the desired spatial configuration. The main difference lies in that Problem $\mathcal{P}3$ requires less information than Problem $\mathcal{P}2$, consuming less communication resource. Figs. 6-13 are the simulation results by using Problem $\mathcal{P}3$. Fig. 6 shows the actual trajectory of the three quadrotors. it can be seen that the three agents track the straight line while maintaining the formation. Fig. 7 indicates that the leader is able to track the reference trajectory. Fig. 8 shows that the control input constraints of the leader are not violated. Figs. 9-10 are the relative distance among the followers and leader, implying that the desired spatial configuration is formed. For ease of checking the satisfaction of constraints, the control input signal is transferred into the rotor's angular velocity with (4). Figs. 11-12 show the detailed angular velocity values applied on the followers, which satisfy the constraint in (26). We use the calculation time interval of OCP to denote the computation cost. Fig.13 shows that computation burdens of the followers have been greatly decreased. Figs. 14-16 are the simulation results by using Problem $\mathcal{P}2$. We can see from Fig. 14 that the trajectories of the follower agents are smoother than Fig. 6. In Figs. 15-16, the predefined formation is reached in 2.5s, which is shorter than the one in Figs. 9-10. This is reasonable because that the MPC controller in the latter uses more information, which helps to improve the accuracy of the control scheme.

B. EXAMPLE TWO

In this example, we consider the formation control for the multi-quadrotors, where the reference trajectory for the leader

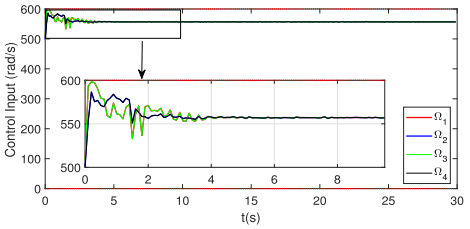

FIGURE 12. The control input of the follower 2 with two-layer model (line).

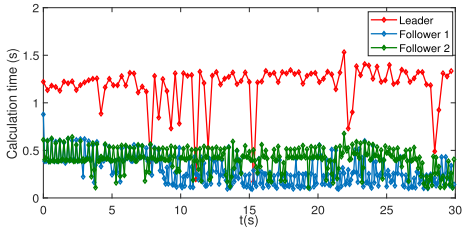

FIGURE 13. Calculation time of the three quadrotors.

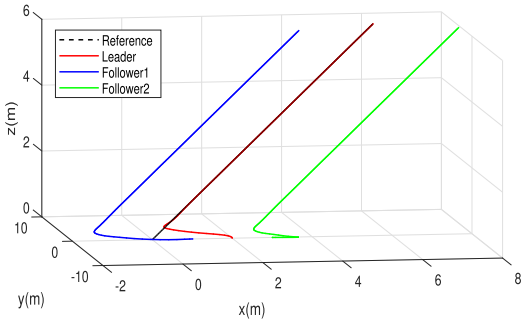

FIGURE 14. Trajectory of the three quadroters under sufficient information (line).

is a spiral ascending curve formulated by

$$\begin{aligned} x_r(t) &= 10\cos(t/16), \\ y_r(t) &= 10\sin(t/16), \\ z_r(t) &= 0.1t, \\ \phi_r(t) &= 0, \theta_r(t) = 0, \psi_r(t) = \pi/2 + 0.001t. \end{aligned} \tag{28}$$

The initial system state $\boldsymbol{x}_l(0)$ for the leader is $[4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]^T$. For the two followers, the initial system states in the upper level are set to be $\boldsymbol{x}_{tr}^1(0) = [3, 0, 0, 0, 0, 0]^T$ and $\boldsymbol{x}_{tr}^2(0) = [4.5, 0, 0, 0, 0, 0]^T$, respectively. $\boldsymbol{x}_{ro}^1(0) = [0, 0, 0, 0, 0, 0]^T$ and $\boldsymbol{x}_{ro}^2(0) = [0, 0, 0, 0, 0, 0]^T$ are the initial system state in the lower level. The other parameters relating to the cost functions are the same as the values in example one.

Fig. 17 shows the 3D trajectories of the three quadrotors. Figs. 18-20 give the tracking results of the leader. We can see from Figs. 18-19 that the leader asymptotically converge to the reference spiral ascending curve while following the target attitude. Fig. 20 shows the satisfactory of the leader's input constraints. Controlled by the designed two-layer distributed MPC formation algorithm, Figs. 21-23 and

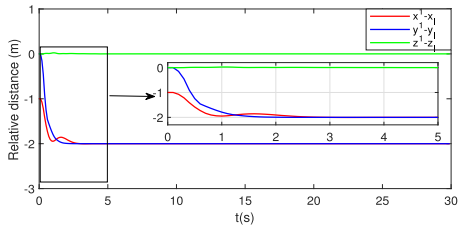

FIGURE 15. Relative distance between the follower 1 (two-layer model) and the leader (nonlinear model) under sufficient information (line).

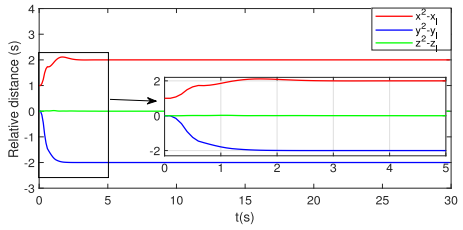

FIGURE 16. Relative distance between the follower 2 (two-layer model) and the leader (nonlinear model) under sufficient information (line).

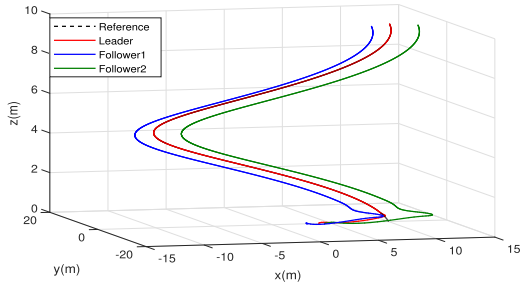

FIGURE 17. Trajectory of the three quadrotors under limited information (curve).

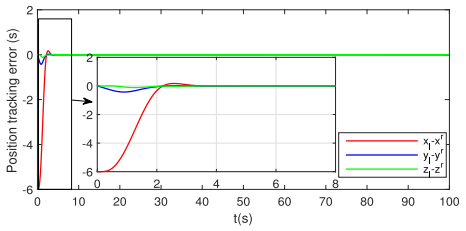

FIGURE 18. Position tracking error of the leader with nonlinear model (curve).

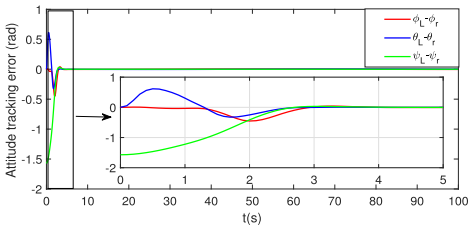

FIGURE 19. Attitude tracking error of the leader with nonlinear model (curve).

Figs. 24-26 are the simulation results of the followers 1 and 2, respectively. Figs. 21-22 and Figs. 24-25 indicate that the

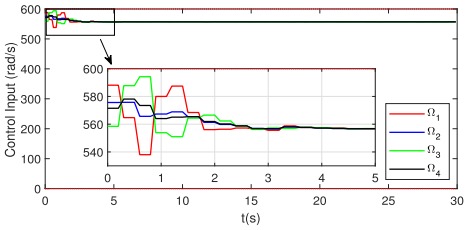

FIGURE 20. The control input of the leader with nonlinear model (curve).

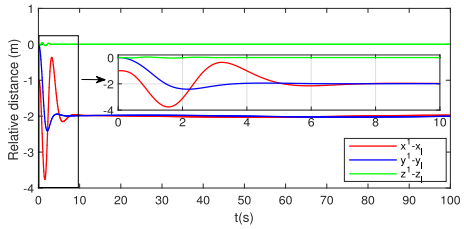

FIGURE 21. Relative distance between the follower 1 (two-layer model) and the leader (nonlinear model) under limited information (curve).

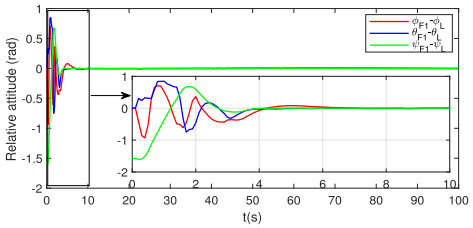

FIGURE 22. Relative attitude between the follower 1 (two-layer model) and the leader (nonlinear model) under limited information (curve).

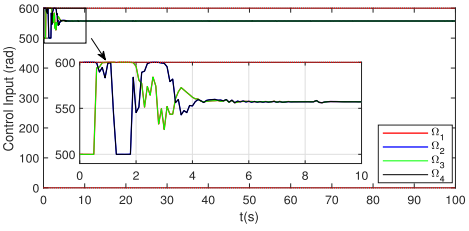

FIGURE 23. The control input of the follower 1 with two-layer model (curve).

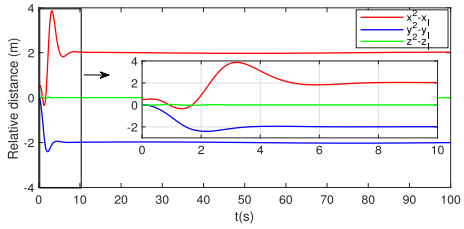

FIGURE 24. Relative distance between the follower 2 (two-layer model) and the leader (nonlinear model) under limited information (curve).

desired distance and attitude error among the agents are reached in 8s, validating the effectiveness of the formation control algorithm. Fig. 23 and Fig. 26 show the rotor's angular

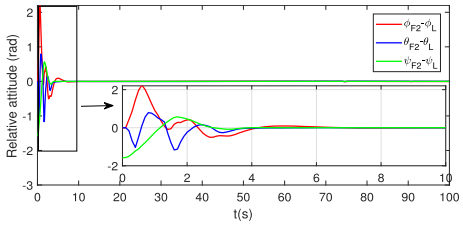

FIGURE 25. Relative attitude between the follower 2 (two-layer model) and the leader (nonlinear model) under limited information (curve).

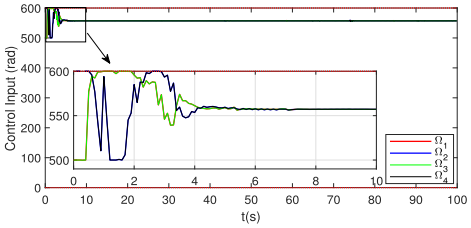

FIGURE 26. The control input of the follower 2 with two-layer model (curve).

velocity of the two followers, which meet the requirements of the angular velocity constraints in (26). Since we solve OCP using limited information of the neighbors, their exist small variation of the system state. However, as stated in III-B, translating more information will put high requirements on the hardware configuration of the agent. Hence, there is a tradeoff between the cost the control performance of the agent.

V. CONCLUSION

In this paper, we have developed an MPC based control strategy to solve the trajectory tracking and formation control problems of the multi-UAVs. For the leader, a new MPC trajectory tracking controller has been designed, which provides a feasible trajectory for the followers. Then, a two-layer distributed MPC formation controller with limited neighbor information has also been developed. Finally, the simulation examples have verified the effectiveness of the proposed algorithms.

The controller in this study is designed using the nominal model. However, due of the following two facts, this model cannot effectively reflect the state evolution process of the actual system: (i)The deviation of the center of gravity from the actual center of mass do exists, resulting in the modeling inaccuracy. (ii) The external disturbances caused by actuator faults, sensor faults and climatic circumstances are not considered. The disadvantages listed above motivate us to investigate robust MPC algorithms in our future work. Possible scheme by incorporating the neural network learning system used in [34], [35], and [36] into MPC will be investigated to mitigate for the model uncertainties.

REFERENCES

[1] A. Mirzaeinia, M. Hassanalian, K. Lee, and M. Mirzaeinia, "Energy conservation of V-shaped swarming fixed-wing drones through position reconfiguration," *Aerosp. Sci. Technol.*, vol. 94, Nov. 2019, Art. no. 105398.

[2] T.-M. Nguyen, Z. Qiu, T. H. Nguyen, M. Cao, and L. Xie, "Persistently excited adaptive relative localization and time-varying formation of robot swarms," *IEEE Trans. Robot.*, vol. 36, no. 2, pp. 553–560, Apr. 2020.

[3] Z. Miao, Y.-H. Liu, Y. Wang, G. Yi, and R. Fierro, "Distributed estimation and control for leader-following formations of nonholonomic mobile robots," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 4, pp. 1946–1954, Oct. 2018.

[4] E. Cetinsoy, S. Dikyar, C. Hancer, K. T. Oner, E. Sirimoglu, M. Unel, and M. F. Aksit, "Design and construction of a novel quad tilt-wing UAV," *Mechatronics*, vol. 22, no. 6, pp. 723–745, Sep. 2012.

[5] T. R. H. Goodbody, N. C. Coops, P. Tompalski, P. Crawford, and K. J. K. Day, "Updating residual stem volume estimates using ALS- and UAV-acquired stereo-photogrammetric point clouds," *Int. J. Remote Sens.*, vol. 38, nos. 8–10, pp. 2938–2953, May 2017.

[6] T. Patterson, S. I. McClean, G. P. Parr, P. J. Morrow, L. Teacy, and J. Nie, "Integration of terrain image sensing with UAV safety management protocols," in *Sensor Systems and Software*. New York, NY, USA: Springer-Verlag, 2010, pp. 36–51.

[7] T. Oliveira, A. P. Aguiar, and P. Encarnação, "Moving path following for unmanned aerial vehicles with applications to single and multiple target tracking problems," *IEEE Trans. Robot.*, vol. 32, no. 5, pp. 1062–1078, Oct. 2016.

[8] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, Mar. 2015.

[9] H. Wang and M. Chen, "Trajectory tracking control for an indoor quadrotor UAV based on the disturbance observer," *Trans. Inst. Meas. Control*, vol. 38, no. 6, pp. 675–692, Jun. 2016.

[10] Y. Alothman, M. Guo, and D. Gu, "Using iterative LQR to control two quadrotors transporting a cable-suspended load," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 4324–4329, Jul. 2017.

[11] X.-L. Lin, C.-F. Wu, and B.-S. Chen, "Robust H_∞ adaptive fuzzy tracking control for MIMO nonlinear stochastic Poisson jump diffusion systems," *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 3116–3130, Aug. 2019.

[12] E.-H. Zheng, J.-J. Xiong, and J.-L. Luo, "Second order sliding mode control for a quadrotor UAV," *ISA Trans.*, vol. 53, no. 4, pp. 1350–1356, Jul. 2014.

[13] Z. Zuo, "Trajectory tracking control design with command-filtered compensation for a quadrotor," *IET Control Theory Appl.*, vol. 4, no. 11, pp. 2343–2355, Nov. 2010.

[14] A. Mokhtari, A. Benallegue, and Y. Orlov, "Exact linearization and sliding mode observer for a quadrotor unmanned aerial vehicle," *Int. J. Robot. Autom.*, vol. 21, no. 1, pp. 39–49, 2006.

[15] S. Li, Y. Wang, J. Tan, and Y. Zheng, "Adaptive RBFNNs/integral sliding mode control for a quadrotor aircraft," *Neurocomputing*, vol. 216, pp. 126–134, Dec. 2016.

[16] E. Kayacan and R. Maslim, "Type-2 fuzzy logic trajectory tracking control of quadrotor VTOL aircraft with elliptic membership functions," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 1, pp. 339–348, Feb. 2017.

[17] J. Hwangbo, I. Sa, R. Siegwart, and M. Hutter, "Control of a quadrotor with reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 2096–2103, Oct. 2017.

[18] S. Kim, H. Oh, and A. Tsourdos, "Nonlinear model predictive coordinated standoff tracking of a moving ground vehicle," *J. Guid., Control, Dyn.*, vol. 36, no. 2, pp. 557–566, Mar. 2013.

[19] F. Gavilan, R. Vazquez, and S. Esteban, "Trajectory tracking for fixed-wing UAV using model predictive control and adaptive backstepping," *IFAC-PapersOnLine*, vol. 48, no. 9, pp. 132–137, 2015.

[20] I. Prodan, S. Olaru, R. Bencatel, J. B. de Sousa, C. Stoica, and S.-I. Niculescu, "Receding horizon flight control for trajectory tracking of autonomous aerial vehicles," *Control Eng. Pract.*, vol. 21, no. 10, pp. 1334–1349, Oct. 2013.

[21] M. Islam, M. Okasha, and E. Sulaeman, "A model predictive control (MPC) approach on unit quaternion orientation based quadrotor for trajectory tracking," *Int. J. Control, Autom. Syst.*, vol. 17, no. 11, pp. 2819–2832, Nov. 2019.

[22] Q. Xu, Z. Wang, and Y. Li, "Fuzzy adaptive nonlinear information fusion model predictive attitude control of unmanned rotorcrafts," *Aerosp. Sci. Technol.*, vol. 98, Mar. 2020, Art. no. 105686.

[23] X. Peng, Z. Sun, K. Guo, and Z. Geng, "Mobile formation coordination and tracking control for multiple nonholonomic vehicles," *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 3, pp. 1231–1242, Jun. 2020.

[24] X. Dong, B. Yu, Z. Shi, and Y. Zhong, "Time-varying formation control for unmanned aerial vehicles: Theories and applications," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 1, pp. 340–348, Jan. 2015.

[25] D. Zhou and M. Schwager, "Virtual rigid bodies for coordinated agile maneuvering of teams of micro aerial vehicles," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 1737–1742.

[26] D. Huang, H. Li, and X. Li, "Formation of generic UAVs-USVs system under distributed model predictive control scheme," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 12, pp. 3123–3127, Dec. 2020.

[27] H. Liu, J. Hu, C. Zhao, X. Hou, Z. Xu, and Q. Pan, "Consensus-based control of multiple fixed-wing UAVs using distributed model predictive control," in *Proc. 7th Int. Conf. Inf., Cybern., Comput. Social Syst. (ICCSS)*, Guangdong, China, Nov. 2020, pp. 858–863.

[28] Z. Cai, H. Zhou, J. Zhao, K. Wu, and Y. Wang, "Formation control of multiple unmanned aerial vehicles by event-triggered distributed model predictive control," *IEEE Access*, vol. 6, pp. 55614–55627, 2018, doi: 10.1109/ACCESS.2018.2872529.

[29] P. Wang and B. Ding, "Distributed RHC for tracking and formation of nonholonomic multi-vehicle systems," *IEEE Trans. Autom. Control*, vol. 59, no. 6, pp. 1439–1453, Jun. 2014.

[30] S. Bouabdallah, P. Murrieri, and R. Siegwart, "Design and control of an indoor micro quadrotor," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, New Orleans, LA, USA, Apr./May 2004, pp. 4393–4398.

[31] Q. Xu, Z. Wang, and Z. Zhen, "Information fusion estimation-based path following control of quadrotor UAVs subjected to Gaussian random disturbance," *ISA Trans.*, vol. 99, pp. 84–94, Apr. 2020.

[32] P. Duan, K. Liu, N. Huang, and Z. Duan, "Event-based distributed tracking control for second-order multiagent systems with switching networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 9, pp. 3220–3230, Sep. 2020.

[33] P. Duan, Q. Wang, Z. Duan, and G. Chen, "A distributed optimization scheme for state estimation of nonlinear networks with norm-bounded uncertainties," *IEEE Trans. Autom. Control*, vol. 67, no. 5, pp. 2582–2589, May 2022.

[34] Z. Yu, Y. Zhang, B. Jiang, X. Yu, J. Fu, Y. Jin, and T. Chai, "Distributed adaptive fault-tolerant close formation flight control of multiple trailing fixed-wing UAVs," *ISA Trans.*, vol. 106, pp. 181–199, Nov. 2020.

[35] Z. Yu, Y. Zhang, B. Jiang, C.-Y. Su, J. Fu, Y. Jin, and T. Chai, "Fractional-order adaptive fault-tolerant synchronization tracking control of networked fixed-wing UAVs against actuator-sensor faults via intelligent learning mechanism," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 12, pp. 5539–5553, Dec. 2021.

[36] Z. Yu, Y. Zhang, B. Jiang, C.-Y. Su, J. Fu, Y. Jin, and T. Chai, "Enhanced recurrent fuzzy neural fault-tolerant synchronization tracking control of multiple unmanned airships via fractional calculus and fixed-time prescribed performance function," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 10, pp. 4515–4529, Oct. 2022.

TIANLAI XU received the B.Sc., M.Sc., and Ph.D. degrees in aeronautical and astronautical science and technology from the Harbin Institute of Technology, Harbin, China, in 2001, 2003, and 2008, respectively. He is currently an Associate Professor with the School of Astronautics, Harbin Institute of Technology. His research interests include aircraft dynamics and control, autonomous navigation, and data fusion.

JINLONG LIU received the B.Sc. degree from Harbin Engineering University, Harbin, China, in 2017, and the M.Sc. degree from the Harbin Institute of Technology, Harbin, in 2019, where he is currently pursuing the Ph.D. degree in aeronautical and astronautical science and technology. His current research interests include multi-UAV task allocation, trajectory planning, and formation control.

ZEXU ZHANG received the B.Sc. degree in electronics and communication engineering from Air Force Engineering University, Xi'an, China, in 1993, and the M.Sc. degree in astronautics engineering and the Ph.D. degree in aerospace science and technology from the School of Astronautics, Harbin Institute of Technology, Harbin, China, in 2000 and 2004, respectively.

He is currently a Full Professor with the Harbin Institute of Technology, where he is also the Director of the Institute of Aircraft Dynamics and Control. His research interests include aircraft autonomous navigation and control, intelligent cooperative perception, and autonomous decision-making in drone swarm and data visualization.

Prof. Zhang was the Elected Director of the Committee of Space Intelligence of the Chinese Society of Space Research, from 2021 to 2025.

GUODONG CHEN received the B.Sc. and M.Sc. degrees from Shenyang Aerospace University, Shenyang, China, in 2017 and 2020, respectively. He is currently pursuing the Ph.D. degree in aeronautical and astronautical science and technology with the Harbin Institute of Technology, China. His current research interests include UAV-swarm trajectory planning and model predictive control.

DI CUI received the B.Sc. and M.Sc. degrees from Northwestern Polytechnical University, Xi'an, China, in 2017 and 2020, respectively, where she is currently pursuing the Ph.D. degree in automatic control. Her current research interests include model predictive control and distributed optimization.

HUIPING LI (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in mechanical engineering and automatic control from Northwestern Polytechnical University, Xi'an, China, and the Ph.D. degree in mechanical engineering from the University of Victoria, Victoria, BC, Canada. He is currently a Full Professor with the School Marine Science and Technology, Northwestern Polytechnical University. His research interests include model predictive control, multi-agent systems, and distributed and cooperative control of unmanned vehicle systems. He serves as an Associate Editor for IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS and *ASME Journal on Dynamic Systems, Measurement and Control*. He was a Guest Technical Editor for a Special Section in IEEE/ASME TRANSACTIONS ON MECHATRONICS.