

Received 20 November 2023, accepted 3 December 2023, date of publication 7 December 2023, date of current version 21 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3340417

## RESEARCH ARTICLE

# Consensus-Based Distributed Formation Control of Multi-Quadcopter Systems: Barrier Lyapunov Function Approach

NARGESS SADEGHZADEH-NOKHODBERIZ<sup>1</sup> AND NADER MESKIN<sup>2</sup>, (Senior Member, IEEE)

<sup>1</sup>Department of Electrical and Computer Engineering, Qom University of Technology, Qom 37181-46645, Iran

<sup>2</sup>Department of Electrical Engineering, Qatar University, Doha, Qatar

Corresponding author: Nader Meskin (nader.meskin@qu.edu.qa)

This work was supported by the Open Access funding provided by the Qatar National Library.

**ABSTRACT** The problem of formation tracking control for a group of quadcopters with nonlinear dynamics using Barrier Lyapunov Functions (BLFs) is studied in this paper where the quadcopters are following a desired predefined trajectory in a predefined formation shape. The BLFs are employed to formulate the problem of formation trajectory tracking with a predefined accuracy. For this purpose, logarithmic BLFs including both the trajectory errors and the errors between the quadcopters' distances with the desired ones (for the formation goal) are proposed. The method is firstly developed in a centralized scheme and then extended to a distributed framework using appropriate asymptotically convergent consensus algorithms. Therefore, the asymptotic convergence of the designed distributed algorithm to the centralized one is guaranteed. Moreover, due to the under-actuated feature of a quadcopter system, a general hierarchical scheme is considered for designing the controller. To this end, firstly a formation altitude tracking control is designed and then using the generated control signal, the formation translational tracking control is developed with the assumption of virtual inputs which are then employed to generate desired trajectory signals for the attitude control subsystem. Finally, attitude controllers are designed separately for each agent using the generated desired signals through logarithmic BLFs to consider a predefined accuracy. The efficiency of the proposed method is demonstrated through simulations and comparisons with the similar approaches in MATLAB-Simulink environment.


**INDEX TERMS** Formation tracking control, multi-quadcopter system, Barrier Lyapunov function (BLF), consensus algorithms.

## I. INTRODUCTION

Unmanned aerial vehicles (UAVs), such as quadcopters, are extensively used in many applications such as military, search and rescue, surveillance, monitoring and navigation in highly dangerous environments and especially in applications with extreme environments where humans cannot access [1], [2], [3]. However, in practice, it is required for a group of UAVs to work with each other to handle a complex mission effectively. For example, it is almost impossible for a single quadcopter to search a vast area. Missions performed by a group of UAVs (generally robots) are more reliable,

time-efficient, and cost-reducing [4]. When a group of UAVs works together, they should travel individually while collaborating with each other to achieve a predefined task. For this purpose, a formation control scheme generates the appropriate control signals for a UAVs' formation to complete their mission.

The problem of formation control in multi-robot systems (both aerial and terrestrial ones) is the main attention of many recent research works [5], [6]. The proposed algorithms can be categorized into two general centralized and decentralized (distributed) schemes where different formation mechanisms of leader-follower [7], virtual structure [8], and behavior-based [9] can be implemented [4]. In order to design a distributed scheme for each of the aforementioned structure,

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

consensus algorithms [10], [11], [12] can be employed to coordinate the agents for the formation control purpose. Such approaches are normally named as consensus-based formation control methods [13], [14], [15], [16], [17]. In the consensus-based formation control, the main purpose is coordination of the agents through consensus algorithms to reach to a given formation. The formation control can also be designed such that agents achieve a predefined shape or formation through tracking a predefined trajectory [18]. Moreover, from the perspective of the sensed and controlled variables, the formation control can be implemented in position-based, displacement-based, and distance-based control schemes [19].

The problem of formation control in multi-quadcopter systems has been studied in some research works more recently. In [20], the problem of finite-time formation control of multi-quadcopter systems in a leader-follower structure is investigated, where the finite-time stability of the proposed distributed controller is shown and the proposed approach guarantees the formation trajectory control using the Lyapunov theory. In [21], a real-time nonlinear model predictive control (NMPC) based leader-follower formation trajectory control of quadcopters are tested on a fleet of CrazyFlie 2.0 quadcopters. The main contribution of the paper is to show that a C++ implementation of MPC as a controller, and not a planner, for real-time formation control of several quadcopters, is feasible. In [22], a model reference adaptive control (MRAC) based leader-follower formation producing problem is studied for a group of quadcopters, where the relative position in the  $x$ - $y$  plane and the relative heading angles of the agents in the presence of disturbances are considered. In [23], a consensus-based leader-follower based formation tracking control for multi-quadcopter systems is presented where the impacts of perturbations are eliminated using an adaptive technique and definition of virtual control signals.

Barrier Lyapunov Functions (BLF) are basically bounded scalar positive definite functions on a given open region defined with respect to a system and approach to infinity on the boundary of that region. They are normally used to design a controller for a system such that state or output constraints are not violated [24]. BLFs in single agent systems are mostly employed to address constraints in the states, especially in the transient time operation. Using BLFs for the formation control of multi-agent systems can be considered as a relative and appropriate idea as, according to [19], multiple agents are driven to achieve prescribed constraints on their states for the formation control. One of the main advantages of using BLFs in the formation control scenario is its ability to define a predefined accuracy for the controller which can help not only for addressing constraints on the states but also the ones caused by disturbances. As a result, more recently, the BLF framework is used for the formation tracking control in multi-robot systems where both constraints of the tracking errors (between the real and the desired trajectories for each robot) and the inter-robot distances are considered for a

multi-robot system in 2D space (3 degrees of freedom) [25]. The proposed method is basically distributed and exponential convergence is guaranteed on the distance tracking errors.

BLFs have been employed for the trajectory control of single quadcopters under state constraints more recently [26], [27], [28], [29]. For example, in [26] the problem of 3-D trajectory following for a nonlinear under-actuated quadcopter dynamics is proposed in which constraints are imposed on its states using BLFs. The constraints include singularity in Euler angles and a predefined accuracy in the trajectory following. However, in spite of many existing research works in formation control of multi-quadcopter systems, utilizing BLFs for this purpose and in multi-quadcopter systems has not been considered in the literature.

Toward this, in this paper, the BLFs are employed for the formation tracking control of multi-quadcopter systems. This goal is implemented through the steps of the backstepping control design framework [30], [31]. The method is firstly developed in a centralized scheme and then extended to a distributed framework using appropriate consensus algorithms. Therefore, as the consensus algorithms asymptotically converge to the desired values, the performance of the proposed distributed method asymptotically converges to the centralized one. Logarithmic BLFs are proposed for our purpose in which both the trajectory errors and the errors between the agents' distances and the desired distances for the formation of the agents are included using the Laplacian of the connectivity graph. Since the quadcopter system is under-actuated, a hierarchical method is employed where first a formation control method is developed for the altitude subsystem and then using the generated control signal, the formation control is designed for the translational subsystem with virtual inputs which are then employed to generate desired trajectory signals for the attitude control subsystem.

The key novelty of our proposed method is by utilizing BLFs in the formation tracking control of multi-quadcopter systems, one can impose prescribed accuracy constraints on the formation control as well as predefined tracking accuracy. The contributions of this paper are summarized as follows:

(1) A centralized hierarchical formation tracking control is designed for the altitude and translational subsystems of a multi-quadcopter system using a proposed logarithmic BLFs in a backstepping procedure.

(2) The designed controllers are extended to distributed ones using asymptotically convergent consensus algorithms.

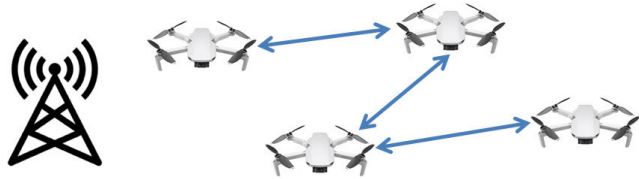
(3) The attitude controllers are designed separately for each agent using the generated desired signals via BLFs to consider a predefined accuracy and the state constraints.

The rest of this paper is organized as follows. In Section II, the problem of multi-quadcopter formation control is introduced and necessary preliminaries are presented. The proposed centralized position control of multi-quadcopter system is derived in Section III. The proposed centralized controller is then transformed to a distributed one in Section IV. The attitude controllers for each agent is designed

in Section V. Simulation results are presented in Section VI. Finally, Section VII provides conclusions and future work.

**II. PRELIMINARIES AND PROBLEM STATEMENT**

In this paper, a group of quadcopters is considered which are communicating with each other through a wireless communication network as depicted in Figure 1. This framework can be modeled using a connectivity graph considering each quadcopter (agent) as a node and each communication as an edge of the graph.



**FIGURE 1.** A group of quadcopters communicating through a wireless communication network.

**A. GRAPH THEORY AND CONSENSUS PROBLEM**

Let  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  be an undirected graph with a non-negative adjacency matrix  $\mathcal{A} = [a_{ij}]$  specifying the interconnection topology of a multi-agent (quadcopter) system where  $\mathcal{V} = \{v_1, \dots, v_N\}$  and  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$  are the set of nodes and edges, respectively. Besides, let  $\mathcal{N}_i = \{v_j \in \mathcal{V} : a_{ij} \neq 0\}$  denotes the set of neighbors of node  $i$ , and  $\mathcal{L}$  is the corresponding Laplacian matrix [32].

*Remark 1:* It is assumed that the connectivity graph of the quadcopters (related to the quadcopters connection topology) is a connected undirected graph.

Now, let  $q_i(t) \in \mathbb{R}^m$  denote the value of node  $v_i$  which represents the  $i$ -agent state vector of a high-pass consensus filter. The consensus filter is a dynamical system with the input and output vectors  $r_i(t) \in \mathbb{R}^m$  and  $l_i(t) \in \mathbb{R}^m$ , respectively, for  $i$ -th node, with the following dynamical model:

$$\begin{aligned} \dot{q}_i(t) &= \sum_{j \in \mathcal{N}_i} a_{ij}(q_j(t) - q_i(t)) + \sum_{j \in \mathcal{N}_i} a_{ij}(r_j(t) - r_i(t)), \\ l_i(t) &= q_i(t) + r_i(t), \end{aligned} \tag{1}$$

Let  $q(t) = [q_1^T(t), \dots, q_N^T(t)]^T$ ,  $r(t) = [r_1^T(t), \dots, r_N^T(t)]^T$  and  $l(t) = [l_1^T(t), \dots, l_N^T(t)]^T$ . Then, the augmented consensus system of (1) can be expressed as follows [33]:

$$\begin{aligned} \dot{q}(t) &= -\hat{L}q(t) - \hat{L}r(t), \\ l(t) &= q(t) + r(t), \end{aligned} \tag{2}$$

where  $\hat{L} = L \otimes I_m$  and  $\otimes$  refers to the Kronecker product and  $I_m$  is an  $m$ -dimensional identity matrix. It is worth mentioning that through computation of the improper MIMO transfer function of the consensus filter presented in (2), it can be easily shown that it is a high-pass consensus filter. Using this high-pass consensus filter in a connected undirected graph, the asymptotic convergence of  $l_i(t)$  to  $\frac{1}{N} \sum_{i=1}^N r_i(t)$  is guaranteed as  $t \rightarrow \infty$  [34].

**B. BARRIER LYAPUNOV THEORY**

To track a desired trajectory with a predefined accuracy, Barrier Lyapunov Functions (BLF) are employed in this paper. Consider the following continuous-time nonlinear system:

$$\dot{x}(t) = f(x(t)), \tag{3}$$

where  $x(t) \in \mathbb{R}^n$  is the system state vector and  $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is the system nonlinear vector function. Then, according to [24], a BLF is a scalar continuous positive definite function  $V : \mathcal{D} \rightarrow \mathbb{R}$ ,  $\mathcal{D} \subset \mathbb{R}^n$  with respect to the system (3) which has continuous first-order partial derivatives at every point of  $\mathcal{D}$  and  $V(x) \rightarrow \infty$  as  $x$  approaches the boundary of  $\mathcal{D}$ . Moreover,  $V(x(t)) \leq b$ ,  $\forall t \geq 0$  along the solution of (3) for  $x(0) \in \mathcal{D}$  with a positive constant  $b$ .

Now, consider a nonlinear system as:

$$\begin{aligned} \dot{x}_1(t) &= f_1(x_1(t)) + g_1(x_1(t))x_2(t), \\ \dot{x}_2(t) &= f_2(x_1(t), x_2(t)) + g_2(x_1(t), x_2(t))u(t), \end{aligned} \tag{4}$$

where  $x_1(t) \in \mathbb{R}^{n_1}$ ,  $x_2(t) \in \mathbb{R}^{n_2}$  are the system states,  $u(t) \in \mathbb{R}^{n_2}$  is the system input, and  $f_1, g_1, f_2$ , and  $g_2$  are smooth vector functions. The goal is  $x_1(t)$  to follow the desired trajectory  $x_{1d}(t)$  with a predefined accuracy. In other words, if one lets  $e_1(t) := x_1(t) - x_{1d}(t)$ , then the goal is  $|e_{1i}(t)| < k_{b1}$ ,  $t \geq 0$ ,  $i = 1, \dots, N$ , where  $k_{b1}$  is a positive predefined scalar. By extension of the idea of using BLF in a backstepping procedure as presented in [24], [31], [35] to the vector form case, the following BLF can be considered for (4):

$$\begin{aligned} V_1(e_1(t)) &= \frac{1}{2} \log \frac{k_{b1}^2}{k_{b1}^2 - e_1^T(t)e_1(t)}, \\ V(e_1(t), e_2(t)) &= V_1(e_1(t)) + V_2(e_2(t)), \end{aligned} \tag{5}$$

where  $V_2(e_2(t)) = \frac{1}{2}e_2^T(t)e_2(t)$  with  $e_2(t) := x_2(t) - \alpha(t)$  and  $\alpha(t)$  is a stabilizing vector function to be designed. According to Lemma 1 in [24] if the inequality of  $\dot{V}(e_1(t), e_2(t)) \leq 0$  holds  $\forall t \geq 0$ , then  $|e_{1i}(t)| < k_{b1}$ ,  $t \geq 0$  if  $\|e_1(0)\| < k_{b1}$ .

**C. QUADCOPTER MODEL**

A quadrotor UAV includes four rotors that generate propeller forces. Schematic of the quadcopter with body coordinate system is depicted in Figure 2. Variations on forces ( $f_1$  to  $f_4$ ) and moments ( $\tau_{M1}$  to  $\tau_{M4}$ ) by adjusting rotors' speeds ( $\omega_1$  to  $\omega_4$ ) produce attitude and translation change in the quadrotor [1], [36]. Let a group of quadcopters (as depicted in Figure 1) consisting of  $N$  quadcopters communicating with each other constructs a connected undirected connectivity graph. Consider the index  $i$  for the  $i$ -th quadcopter parameters, i.e.  $f_{1i}$  to  $f_{4i}$ ,  $\tau_{M1i}$  to  $\tau_{M4i}$  and  $\omega_{1i}$  to  $\omega_{4i}$ .

The quadrotor attitude dynamics for the  $i$ -th quadcopter,  $i = 1, \dots, N$  can be written in the following form:

$$\begin{aligned} \dot{p}_i(t) &= a_{1i}q_i(t)r_i(t) - a_{2i}\Omega_{ri}(t)q_i(t) + b_{1i}u_{2i}(t), \\ \dot{q}_i(t) &= a_{3i}p_i(t)r_i(t) + a_{4i}\Omega_{ri}(t)p_i(t) + b_{2i}u_{3i}(t), \\ \dot{r}_i(t) &= a_{5i}p_i(t)q_i(t) + b_{3i}u_{4i}(t), \end{aligned} \tag{6}$$

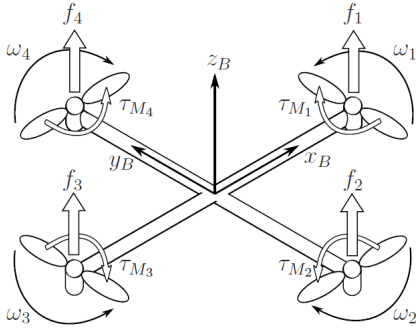


FIGURE 2. Schematic of quadrotor with coordinate axes [36].

where  $p_i(t)$ ,  $q_i(t)$ , and  $r_i(t)$  are angular velocities rotating around  $x$ -axis,  $y$ -axis, and  $z$ -axis in the body frame, respectively, the input signals  $u_{2i}(t)$ ,  $u_{3i}(t)$ , and  $u_{4i}(t)$  are torques in the direction of the corresponding body frame angles. It is also worth mentioning that  $u_{2i}(t) = L_i k_i (-\omega_{2i}^2(t) + \omega_{4i}^2(t))$ ,  $u_{3i}(t) = L_i k_i (-\omega_{1i}^2(t) + \omega_{3i}^2(t))$ , and  $u_{4i}(t) = \sum_{l=1}^4 \tau_{Mli}(t)$ , where  $k_i$  is the lift constant and  $L_i$  is the length between center of the mass and the rotor axes. Moreover,  $a_{1i} = \frac{I_{yyi} - I_{zzi}}{I_{xxi}}$ ,  $a_{2i} = \frac{I_{ri}}{I_{xxi}}$ ,  $b_{1i} = \frac{L_i}{I_{xxi}}$ ,  $a_{3i} = \frac{I_{zzi} - I_{xxi}}{I_{yyi}}$ ,  $a_{4i} = \frac{I_{ri}}{I_{yyi}}$ ,  $b_{2i} = \frac{L_i}{I_{yyi}}$ ,  $a_{5i} = \frac{I_{xxi} - I_{yyi}}{I_{zzi}}$ ,  $b_{3i} = \frac{L_i}{I_{zzi}}$ , with the inertia tensors represented as  $I_{xxi}$ ,  $I_{yyi}$  and  $I_{zzi}$ ,  $I_{ri}$  is the inertia of the propellers,  $\Omega_{ri}(t) = \omega_{1i}(t) - \omega_{2i}(t) + \omega_{3i}(t) - \omega_{4i}(t)$  describes the relative speed of propeller.

The relationship between angular velocities in the body and inertial frames are described as follows:

$$\begin{aligned} \dot{\phi}_i(t) &= p_i(t) + \sin(\phi_i(t)) \tan(\theta_i(t)) q_i(t) \\ &\quad + \cos(\phi_i(t)) \tan(\theta_i(t)) r_i(t), \\ \dot{\theta}_i(t) &= \cos(\theta_i(t)) q_i(t) - \sin(\theta_i(t)) r_i(t), \\ \dot{\psi}_i(t) &= \frac{\sin(\phi_i(t))}{\cos(\theta_i(t))} q_i(t) + \frac{\cos(\phi_i(t))}{\cos(\theta_i(t))} r_i(t), \end{aligned} \quad (7)$$

where the roll angle  $\phi_i(t)$ , pitch angle  $\theta_i(t)$ , and yaw angle  $\psi_i(t)$  determine rotations around  $x$ -axis,  $y$ -axis, and  $z$ -axis, respectively. Finally, the quadcopter translational dynamics is presented in the following:

$$\begin{aligned} \ddot{x}_i(t) &= \frac{u_{1i}(t)}{m_i} (\cos(\psi_i(t)) \sin(\theta_i(t)) \cos(\phi_i(t)) \\ &\quad + \sin(\psi_i(t)) \sin(\phi_i(t))), \\ \ddot{y}_i(t) &= \frac{u_{1i}(t)}{m_i} (\sin(\psi_i(t)) \sin(\theta_i(t)) \cos(\phi_i(t)) \\ &\quad - \cos(\psi_i(t)) \sin(\phi_i(t))), \\ \ddot{z}_i(t) &= -g + \frac{u_{1i}(t)}{m_i} \cos(\phi_i(t)) \cos(\theta_i(t)), \end{aligned} \quad (8)$$

where  $[x_i(t) \ y_i(t) \ z_i(t)]^T$  represents the position of quadcopter in the inertial frame,  $u_{1i}(t)$  defines the main thrust created by combined forces of rotors, that is  $u_{1i}(t) = \sum_{l=1}^4 f_{li}(t)$ ,  $g$  is the gravity constant, and  $m_i$  is the mass of the  $i$ -th quadcopter.

Now, in the rest of the paper, it is intended to solve the following problems:

*Problem 1:* How to design a centralized controller to control the position (altitude and translation) of a group of quadcopters to follow a desired trajectory in a predefined formation when the position subsystem is under-actuated?

*Problem 2:* How to implement the proposed centralized formation control algorithm in a distributed manner?

*Problem 3:* How to control the attitude subsystems of each agent to track the generated desired trajectory from previous steps with a predefined accuracy?

### III. CENTRALIZED POSITION CONTROL

In this section, to solve **Problem 1**, a centralized controller is designed for the position subsystem. As the position subsystem is under-actuated, at first, the altitude subsystem is controlled and then the translational subsystem is controlled using virtual inputs. It is worth mentioning that, this general hierarchical framework is similar to the one proposed in [26]. However, the main difference is that our proposed method has been extended and developed for the formation tracking control of a multi quadcopter system. In other words, several agents should be able to track the desired trajectory in a formation paradigm.

#### A. SYSTEM DESCRIPTION AND CONTROL GOALS

In this subsection, the centralized model of altitude and translational subsystems are firstly introduced and then the control objectives for trajectory tracking and formation control are presented.

##### 1) SYSTEM DESCRIPTION

The altitude subsystem, as presented in (8), is firstly reformulated in the form of continuous-time state space model for the  $i$ -th agent as:

$$\begin{aligned} \dot{x}_{1i}(t) &= x_{2i}(t), \\ \dot{x}_{2i}(t) &= -g + \frac{u_{1i}(t)}{m_i} \cos(\phi_i(t)) \cos(\theta_i(t)), \end{aligned} \quad (9)$$

where  $x_{1i}(t) = z_i(t)$  and  $x_{2i}(t) = \dot{z}_i(t)$ . Now, let  $X_1(t) = [x_{11}(t), \dots, x_{1N}(t)]^T$ ,  $X_2(t) = [x_{21}(t), \dots, x_{2N}(t)]^T$  and  $U_1(t) = [u_{11}(t), \dots, u_{1N}(t)]^T$ , then the following centralized model is obtained:

$$\begin{aligned} \dot{X}_1(t) &= X_2(t), \\ \dot{X}_2(t) &= -G + B_2(t)U_1(t), \end{aligned} \quad (10)$$

where  $G = g1_{N \times 1}$  and

$$B_2(t) = \text{diag} \left[ \frac{1}{m_i} \cos(\phi_i(t)) \cos(\theta_i(t)) \right]_{i=1, \dots, N}.$$

It should be noted that in the singular orientation of  $\cos(\phi_i(t)) \cos(\theta_i(t)) = 0$ , the quadrotor altitude control is not possible. Hence, this singular orientation should be avoided in the attitude control.

Now, in order to model the translational subsystem, since the position subsystem is under-actuated, two virtual inputs

are considered for the translational subsystem according to (8) as follows:

$$\begin{aligned} u_{xi}(t) &= \cos(\psi_i(t)) \sin(\theta_i(t)) \cos(\phi_i(t)) \\ &\quad + \sin(\psi_i(t)) \sin(\phi_i(t)), \\ u_{yi}(t) &= \sin(\psi_i(t)) \sin(\theta_i(t)) \cos(\phi_i(t)) \\ &\quad - \cos(\psi_i(t)) \sin(\phi_i(t)). \end{aligned} \quad (11)$$

Reformulating the translational subsystem according to (8) and (11) in the form of the state space model for the  $i$ -th subsystem, one can conclude:

$$\begin{aligned} \dot{x}_{3i}(t) &= x_{4i}(t), \\ \dot{x}_{4i}(t) &= \frac{u_{1i}(t)}{m_i} u_{vi}(t), \end{aligned} \quad (12)$$

where  $x_{3i}(t) = [x_i(t) \ y_i(t)]^T$ ,  $x_{4i}(t) = [\dot{x}_i(t) \ \dot{y}_i(t)]^T$ , and  $u_{vi}(t) = [u_{xi}(t) \ u_{yi}(t)]^T$ . The collective state-space model is then can be formulated as:

$$\begin{aligned} \dot{X}_3(t) &= X_4(t), \\ \dot{X}_4(t) &= B_4(t) \otimes I_2 U_v(t), \end{aligned} \quad (13)$$

where  $\otimes$  refers to the Kronecker product,  $X_3(t) = [x_{31}^T(t), \dots, x_{3N}^T(t)]^T$ ,  $X_4 = [x_{41}^T(t), \dots, x_{4N}^T(t)]^T$ ,  $U_v(t) = [u_{v1}^T(t), \dots, u_{vN}^T(t)]^T$ , and  $B_4(t) = \text{diag}[\frac{u_{1i}(t)}{m_i} |_{i=1, \dots, N}]$ . It should be noted that  $u_{1i}(t) \neq 0, \forall t \geq 0$  as it is assumed that the rotors are not ever turn off during the maneuver and a minimum trust should be generated by the rotors for hovering.

## 2) CONTROL OBJECTIVES

Two different types of control objectives are considered in this paper. The first one is devoted to the tracking problem and the second one is related to the formation control.

Let  $X_{1d}(t) = [x_{11d}(t), \dots, x_{1Nd}(t)]^T$  where  $x_{1id}(t), i = 1, \dots, N$  are continuous-time desired altitude trajectory signals with bounded first and second order derivatives and let  $E_1(t) := X_1(t) - X_{1d}(t)$  where  $E_1(t) = [e_{11}(t), \dots, e_{1N}(t)]^T$ . The control goal for tracking the desired altitude trajectory with a predefined accuracy is to design the control input  $U_1$  such that:

$$|e_{1i}(t)| < k_{b1}, \forall t \geq 0, \quad (14)$$

where  $k_{b1}$  is a positive predefined scalar.

Moreover, let  $X_{3d}(t) = [x_{31d}^T(t), \dots, x_{3Nd}^T(t)]^T$  where  $x_{3id}(t) |_{i=1}^N$  are continuous-time desired translational trajectory vector signals with bounded first and second order derivatives. Besides, let  $E_3(t) := X_3(t) - X_{3d}(t)$  where  $E_3(t) = [e_{31}^T(t), \dots, e_{3N}^T(t)]^T$ . It is supposed to design  $U_v(t)$  such that the following control goals are fulfilled. Firstly, the desired translational trajectory  $X_{3d}(t)$  should be tracked with a predefined accuracy:

$$\|e_{3i}(t)\| < k_{b3}, \forall t \geq 0, i = 1, \dots, N, \quad (15)$$

where  $k_{b3}$  is a positive predefined scalar

Moreover, the following conditions are considered to keep the quadcopters in a desired formation with

a predefined accuracy:

$$\sum_{i,j \in \mathcal{E}} (|x_{1i}(t) - x_{1j}(t)| - |x_{1id}(t) - x_{1jd}(t)|)^2 \leq k_{e1}^2, \quad (16)$$

$$\sum_{i,j \in \mathcal{E}} (\|x_{3i}(t) - x_{3j}(t)\| - \|x_{3di}(t) - x_{3dj}(t)\|)^2 \leq k_{e3}^2, \quad (17)$$

where  $k_{e1}$  and  $k_{e3}$  are positive predefined scalars.

## B. ALTITUDE CONTROL

In this section, the altitude control problem is studied considering both trajectory tracking and formation control as presented in (14) and (16), respectively.

*Theorem 1:* Consider the altitude dynamic (10) with the following control input

$$\begin{aligned} U_1(t) &= B_2^{-1}(t) (\dot{A}_2(t) - \frac{(a_{e1}(t)I_N + a_{b1}(t)\mathcal{L})E_1(t)}{a_{e1}(t)a_{b1}(t)} \\ &\quad - k_2 E_2(t) + G), \end{aligned} \quad (18)$$

where

$$A_2(t) = \dot{X}_{1d}(t) - k_1 a_{b1}(t) a_{e1}(t) (a_{e1}(t)I_N + a_{b1}(t)\mathcal{L})^{-1} E_1(t), \quad (19)$$

with  $k_1$  and  $k_2$  as positive constants. Then, the state and formation constraints (14) and (16) are satisfied  $\forall t > 0$ , if  $\|E_1(0)\| < k_{b1}$  and  $E_1^T(0)\mathcal{L}E_1(0) < k_{e1}^2$ .

*Proof:* First, using the fact that  $(|a| - |b|)^2 \leq (a - b)^2$  it follows that:

$$\begin{aligned} &\sum_{i,j \in \mathcal{E}} (|x_{1i}(t) - x_{1j}(t)| - |x_{1id}(t) - x_{1jd}(t)|)^2 \\ &\leq \sum_{i,j \in \mathcal{E}} (x_{1i}(t) - x_{1j}(t) - x_{1id}(t) + x_{1jd}(t))^2. \end{aligned} \quad (20)$$

Therefore, since  $x_{1i}(t) - x_{1j}(t) - x_{1id}(t) + x_{1jd}(t) = e_{1i}(t) - e_{1j}(t)$ , the condition in (16) can be replaced with the following one:

$$\begin{aligned} &\sum_{i,j \in \mathcal{E}} (|x_{1i}(t) - x_{1j}(t)| - |x_{1id}(t) - x_{1jd}(t)|)^2 \\ &\leq \sum_{i,j \in \mathcal{E}} (e_{1i}(t) - e_{1j}(t))^2 < k_{e1}^2. \end{aligned} \quad (21)$$

Therefore, using the Laplacian matrix properties ( $\mathcal{L}$ ) the following is obtained:

$$E_1^T(t)\mathcal{L}E_1(t) = \sum_{i,j \in \mathcal{E}} (e_{1i}(t) - e_{1j}(t))^2 < k_{e1}^2. \quad (22)$$

Now, the following BLF candidate is chosen:

$$\begin{aligned} V_1(E_1(t)) &= \frac{1}{2} \log \frac{k_{b1}^2}{k_{b1}^2 - E_1^T(t)E_1(t)} \\ &\quad + \frac{1}{2} \log \frac{k_{e1}^2}{k_{e1}^2 - E_1^T(t)\mathcal{L}E_1(t)}. \end{aligned} \quad (23)$$

For the sake of simplicity, let  $a_{b1}(t) := k_{b1}^2 - E_1^T(t)E_1(t)$  and  $a_{e1}(t) := k_{e1}^2 - E_1^T(t)\mathcal{L}E_1(t)$ . Let  $E_2(t) := X_2(t) - A_2(t)$

where  $A_2(t)$  is stabilizing vector parameter defined in (19) and  $E_2(t) = [e_{21}(t), \dots, e_{2N}(t)]^T$ . Therefore:

$$\dot{V}_1(t) = \frac{E_1^T(t)\dot{E}_1(t)}{a_{b1}(t)} + \frac{E_1^T(t)\mathcal{L}\dot{E}_1(t)}{a_{e1}(t)}. \quad (24)$$

Since  $\dot{E}_1(t) = X_2(t) - \dot{X}_{1d}(t) = E_2(t) + A_2(t) - \dot{X}_{1d}(t)$ , (24) can be rewritten as follow:

$$\begin{aligned} \dot{V}_1(t) &= \frac{E_1^T(t)(E_2(t) + A_2(t) - \dot{X}_{1d}(t))}{a_{b1}(t)} \\ &+ \frac{E_1^T(t)\mathcal{L}(E_2(t) + A_2(t) - \dot{X}_{1d}(t))}{a_{e1}(t)} \\ &= \frac{E_1^T(t)(a_{e1}(t)I_N + a_{b1}(t)\mathcal{L})(E_2(t) + A_2(t) - \dot{X}_{1d}(t))}{a_{e1}(t)a_{b1}(t)}, \end{aligned} \quad (25)$$

where  $I_N$  is an  $N \times N$  identity matrix. The selection of  $A_2(t)$  in (19) leads to:

$$\dot{V}_1(t) = \frac{E_1^T(t)(a_{e1}(t)I_N + a_{b1}(t)\mathcal{L})E_2(t)}{A_{e1}(t)A_{b1}(t)} - k_1 E_1^T(t)E_1(t). \quad (26)$$

Now, a Lyapunov function candidate by augmenting  $V_1(E_1(t))$  with a quadratic function is chosen in the next step to cancel out the first term in  $\dot{V}_1(t)$  through an appropriate selection of the control input  $U_1(t)$ , namely:

$$V_2(E_1(t), E_2(t)) = V_1(E_1(t)) + \frac{1}{2}E_2^T(t)E_2(t). \quad (27)$$

Then, it follows that:

$$\dot{V}_2(t) = \dot{V}_1(t) + E_2^T(t)(-G + B_2(t)U_1(t) - \dot{A}_2(t)). \quad (28)$$

Substituting (18) and (26) in (28) leads to:

$$\dot{V}_2(t) = -k_1 E_1^T(t)E_1(t) - k_2 E_2^T(t)E_2(t) < 0, \quad (29)$$

which proves that (14) and (22) holds, according to the barrier Lyapunov theory as presented in II-B,  $\forall t > 0$  provided  $\|E_1(0)\| < k_{b1}$  and  $E_1^T(0)\mathcal{L}E_1(0) < k_{e1}^2$ .  $\square$

*Remark 2:* It should be noted that  $a_{e1}(t)I_N + a_{b1}(t)\mathcal{L}$  is a positive definite matrix due to the fact that  $\mathcal{L}$  is positive semi-definite and  $a_{e1}(t)$  and  $a_{b1}(t)$  keep staying positive as  $E_1(0)$  starts such that  $E_1^T(0)E_1(0) < k_{b1}^2$  and  $E_1^T(0)\mathcal{L}E_1(0) < k_{e1}^2$  and this leads to  $\dot{V}_2(0) < 0$  as follows and this causes  $E_1^T(t)E_1(t) < k_{b1}^2$  and  $E_1^T(t)\mathcal{L}E_1(t) < k_{e1}^2$ ,  $\forall t > 0$ .

### C. TRANSLATIONAL CONTROL

In this section, the goal is to solve the translational control problem considering both trajectory tracking and formation control as presented in (15) and (17), respectively.

*Theorem 2:* Consider the transnational dynamic (13) with the control input

$$\begin{aligned} U_v(t) &= (B_4(t) \otimes I_2)^{-1} (\dot{A}_4(t) \\ &- \frac{a_{e3}(t)I_{2N} + a_{b3}(t)(\mathcal{L} \otimes I_2)}{a_{e3}(t)a_{b3}(t)} E_3(t) - k_4 E_4(t)), \end{aligned} \quad (30)$$

where

$$\begin{aligned} A_4(t) &= \dot{X}_{3d}(t) - \mathcal{K}_3 a_{b3}(t) a_{e3}(t) (a_{e3}(t) I_{2N} \\ &+ a_{b3}(t) (\mathcal{L} \otimes I_2))^{-1} E_3(t), \end{aligned} \quad (31)$$

with  $k_3$  and  $k_4$  as positive constants. Then, the state and formation constraints (15) and (17) are satisfied  $\forall t > 0$ , if  $\|E_3(0)\| < k_{b3}$  and  $E_3^T(0)(\mathcal{L} \otimes I_2)E_3(0) < k_{e3}^2$ .

*Proof:* Using the fact that  $(\|a\| - \|b\|)^2 \leq \|a - b\|^2$ , the following inequality can be concluded:

$$\begin{aligned} \sum_{i,j \in \mathcal{E}} (\|x_{3i}(t) - x_{3j}(t)\| - \|x_{3di}(t) - x_{3dj}(t)\|)^2 \\ \leq \sum_{i,j \in \mathcal{E}} \|x_{3i}(t) - x_{3j}(t) - x_{3di}(t) + x_{3dj}(t)\|^2. \end{aligned} \quad (32)$$

Then, since  $x_{3i}(t) - x_{3j}(t) - x_{3di}(t) + x_{3dj}(t) = e_{3i}(t) - e_{3j}(t)$ , the condition (17) can be replaced with the following one:

$$E_3^T(t)(\mathcal{L} \otimes I_2)E_3(t) = \sum_{i,j \in \mathcal{E}} \|e_{3i}(t) - e_{3j}(t)\|^2 < k_{e3}^2. \quad (33)$$

The following BLF candidate similar to (23) is chosen:

$$V_3(E_3(t)) = \frac{1}{2} \log \frac{k_{b3}^2}{a_{b3}(t)} + \frac{1}{2} \log \frac{k_{e3}^2}{a_{e3}(t)}, \quad (34)$$

where  $a_{b3}(t) = k_{b3}^2 - E_3^T(t)E_3(t)$  and  $a_{e3}(t) = k_{e3}^2 - E_3^T(t)(\mathcal{L} \otimes I_2)E_3(t)$ . Now, let  $E_4(t) := X_4(t) - A_4(t)$  where  $A_4(t)$  is a stabilizing vector parameter. It can be designed applying the same procedure as in (24) and (25) to  $\dot{V}_3(E_3(t))$  which leads to (31). Moreover, similar to (27), a Lyapunov function candidate by augmenting a quadratic function to  $V_3(E_3(t))$  is chosen leading to  $V_4(E_3(t), E_4(t)) = V_3(E_3(t)) + \frac{1}{2}E_4^T(t)E_4(t)$ . Through computation of  $\dot{V}_4(t)$  and by substituting the control input (30), it follows that  $\dot{V}_4(t) = -k_3 E_3^T(t)E_3(t) - k_4 E_4^T(t)E_4(t) < 0$  which proves that (15) and (17) holds, according to the barrier Lyapunov theory as presented in II-B,  $\forall t > 0$  provided  $\|E_3(0)\| < k_{b3}$  and  $E_3^T(0)(\mathcal{L} \otimes I_2)E_3(0) < k_{e3}^2$ .  $\square$

Similar to Remark 2 it can be shown that  $a_{e3}(t)I_{2N} + a_{b3}(t)(\mathcal{L} \otimes I_2)$  is positive definite and accordingly is invertible.

## IV. DISTRIBUTED IMPLEMENTATION OF THE PROPOSED POSITION CONTROL

In this section, **Problem 2** is studied and it is shown how to implement the derived control laws in the previous section, i.e. (18) and (30), in a distributed way in each agent communicating with the neighboring ones.

### A. DISTRIBUTED ALTITUDE CONTROL

At first, consider the designed control law for the altitude subsystem, that is  $U_1(t)$  in (18). The  $i$ -th element of  $U_1(t)$  is the altitude control law for  $i$ -th agent,  $u_{1i}(t)$ , which can be written as:

$$\begin{aligned} u_{1i}(t) &= [B_2(t)]_{ii}^{-1} (\dot{\alpha}_{2i}(t) - \frac{e_{1i}(t)}{a_{b1}(t)} \\ &- \frac{1}{a_{e1}(t)} \sum_{j \in \mathcal{N}_i} (e_{1i}(t) - e_{1j}(t)) - k_2 e_{2i}(t) + g), \end{aligned} \quad (35)$$

where  $\alpha_{2i}(t)$  is the  $i$ -th element of  $A_2(t)$ . It is also worth reminding that  $B_2(t)$  is diagonal matrix and thus only  $[B_2(t)]_{ii}^{-1}$  gets involved in  $u_{1i}(t)$  computation. According to (35), the  $i$ -th agent is aware of its own and its neighboring states, that is  $e_{1i}(t)$ ,  $e_{1j}(t)$ ,  $j \in \mathcal{N}_i$  and  $e_{2i}(t)$ . However, an important issue is how to compute  $\alpha_{2i}(t)$ ,  $a_{b1}(t)$ , and  $a_{e1}(t)$  in a decentralized way. Toward this, it is proposed in this paper to employ consensus algorithms as introduced in II-A.

1) DISTRIBUTED COMPUTATION OF  $a_{b1}(t)$  AND  $a_{e1}(t)$

As previously defined in (23),  $a_{b1}(t) = k_{b1}^2 - E_1^T(t)E_1(t)$  and  $a_{e1}(t) = k_{e1}^2 - E_1^T(t)\mathcal{L}E_1(t)$ , and information of all agents is required to compute them. Toward this, as mentioned earlier, high-pass consensus filter as introduced in II-A is employed where  $E_1^T(t)E_1(t)$  and  $E_1^T(t)\mathcal{L}E_1(t)$  should be computed for  $a_{b1}(t)$  and  $a_{e1}(t)$  computation, respectively. At first, consider that  $E_1^T(t)E_1(t) = \sum_{i=1}^N e_{1i}^2(t)$ . Using (1), we can compute  $\frac{1}{N}E_1^T(t)E_1(t)$  substituting  $r_i(t) = e_{1i}^2(t)$  and  $r_j(t) = e_{1j}^2(t)$  with the corresponding consensus state of  $q_{ia_{b1}}(t)$ . Then simply, the output of the consensus system,  $l_{ia_{b1}}(t)$  is multiplied by  $N$ . In other words,  $a_{b1i}(t) = k_{b1}^2 - Nl_{ia_{b1}}(t)$  which asymptotically converges to  $a_{b1}(t)$ .

Secondly, to compute  $E_1^T(t)\mathcal{L}E_1(t)$ , the following equality is employed:

$$\sum_{i,j=1}^N a_{ij}(x_i - x_j)^2 = 2x^T \mathcal{L}x, \quad (36)$$

where  $x = [x_1, \dots, x_N]^T$  is a  $N \times 1$  vector. Toward the goal, each agent computes firstly  $r_{ia_{e1}}(t) = \sum_{j \in \mathcal{N}_i} (e_{1i}(t) - e_{1j}(t))^2$ ,  $i = 1, \dots, N$ , and also receives the corresponding computed  $r_{ja_{e1}}(t)$  from its neighbors and the values are substituted in (1) with the consensus state of  $q_{ia_{e1}}(t)$ . Again, clearly, the result of  $l_{ia_{e1}}(t)$  should be multiplied by  $\frac{N}{2}$ . In other words,  $a_{e1i}(t) = k_{e1}^2 - \frac{N}{2}l_{ia_{e1}}(t)$  which asymptotically converges to  $a_{e1}(t)$ .

2) DISTRIBUTED COMPUTATION OF  $\alpha_{2i}(t)$

The main issue in computation of  $\alpha_{2i}(t)$  is how to compute the  $i$ -th element of  $(a_{e1}(t)I_N + a_{b1}(t)\mathcal{L})^{-1}E_1(t)$  according to (19) as all of the agents' states ( $e_{1i}(t)$ ,  $i = 1, \dots, N$ ) are getting involved and only neighboring states are available. For this purpose, each agent computes  $a_{be1i}(t) := (a_{e1i}(t)I_N + a_{b1i}(t)\mathcal{L})^{-1}$  and put  $r_i(t) = [a_{be1i}(t)]_{ii}e_{1i}(t)$  and  $r_j(t) = [a_{be1i}(t)]_{ij}e_{1j}(t)$ ,  $j \in \mathcal{N}_i$  in (1) with the consensus state of  $q_{i\alpha_{2i}}(t)$ . Finally, the filter output computed  $l_{i\alpha_{2i}}(t)$  converges to  $\frac{1}{N} \sum_{j=1}^N [a_{be1i}(t)]_{ij}(t)e_{1j}(t)$ . Hence,  $\alpha_{2i}(t)$  can be obtained as:

$$\alpha_{2i}(t) = \dot{x}_{1id}(t) - Nk_1 a_{b1i}(t) a_{e1i}(t) l_{i\alpha_{2i}}(t). \quad (37)$$

Besides, it is also required for each agent  $i$  to incorporate in distributed computation of  $\alpha_{2l}(t)$ ,  $l = 1, \dots, N$ ,  $l \neq i$  through computation of consensus state of  $q_{i\alpha_{2l}}(t)$  for  $l = 1, \dots, N$ ,  $l \neq i$  with inputs of  $r_j(t) = [a_{be1i}(t)]_{lj}e_{1j}(t)$  and  $r_i(t) = [a_{be1i}(t)]_{li}e_{1i}(t)$ . In order to clarify the proposed distributed altitude control algorithm, Table 1 describes the method in details as a pseudo code.

TABLE 1. Pseudo code corresponding to distributed computation of  $a_{b1}$ ,  $a_{e1}$  and  $\alpha_{2i}$  related to the altitude control.

---

For  $i = 1 : N$   
 for  $j \in \mathcal{N}_i$ : receive  $e_{1j}(t)$ ,  $q_{ja_{b1}}(t)$ ,  $r_{ja_{e1}}(t)$ ,  $q_{ja_{e1}}(t)$ ,  
 $q_{j\alpha_{2l}}(t)$ ,  $l = 1, \dots, N$   
**Step 1:** Compute  $a_{b1i}(t)$ :

- Consensus filter for computation of  $\frac{1}{N}E_1^T(t)E_1(t)$ :  
 $\dot{q}_{ia_{b1}}(t) = \sum_{j \in \mathcal{N}_i} a_{ij}(q_{ja_{b1}}(t) - q_{ia_{b1}}(t)) + \sum_{j \in \mathcal{N}_i} a_{ij}(e_{1j}^2(t) - e_{1i}^2(t))$ ,  
 $l_{ia_{b1}}(t) = q_{ia_{b1}}(t) + e_{1i}^2(t)$
- Computation of  $a_{b1i}(t)$ :  
 $a_{b1i}(t) = k_{b1}^2 - Nl_{ia_{b1}}(t)$

**Step 2:** Compute  $a_{e1i}(t)$ :

- Computation of  $r_{ia_{e1}}(t)$ :  
 $r_{ia_{e1}}(t) = \sum_{j \in \mathcal{N}_i} (e_{1i}(t) - e_{1j}(t))^2$
- Consensus filter for computation of  $\frac{2}{N}E_1^T(t)\mathcal{L}E_1(t)$ :  
 $\dot{q}_{ia_{e1}}(t) = \sum_{j \in \mathcal{N}_i} a_{ij}(q_{ja_{e1}}(t) - q_{ia_{e1}}(t)) + \sum_{j \in \mathcal{N}_i} a_{ij}(r_{ja_{e1}}(t) - r_{ia_{e1}}(t))$ ,  
 $l_{ia_{e1}}(t) = q_{ia_{e1}}(t) + r_{ia_{e1}}(t)$
- Computation of  $a_{e1i}(t)$ :  
 $a_{e1i}(t) = k_{e1}^2 - \frac{N}{2}l_{ia_{e1}}(t)$

**Step 3:** Compute  $\alpha_{2i}(t)$

- Compute  $a_{be1i}(t) = (a_{e1i}(t)I_N + a_{b1i}(t)\mathcal{L})^{-1}$
- Consensus filter for computation of  $\frac{1}{N}a_{be1}^{(i,:)}(t)E_1(t)$  (where  $a^{(i,:)}$  refers to the  $i$ -row of the matrix  $a$ )  
 $\dot{q}_{i\alpha_{2i}}(t) = \sum_{j \in \mathcal{N}_i} a_{ij}(q_{j\alpha_{2i}}(t) - q_{i\alpha_{2i}}(t)) +$   
 $\sum_{j \in \mathcal{N}_i} a_{ij}([a_{be1i}(t)]_{ij}e_{1j}(t) - [a_{be1i}(t)]_{ii}e_{1i}(t))$   
 $l_{i\alpha_{2i}}(t) = q_{i\alpha_{2i}}(t) + [a_{be1i}(t)]_{ii}(t)e_{1i}(t)$
- computation of  $\alpha_{2i}(t)$  using (37)

**Step 4:** Compute  $u_{1i}(t)$  using (35).  
**Step 5:** Compute  $q_{i\alpha_{2l}}(t)$  for  $l = 1 : N$ ,  $l \neq i$ :

$$\dot{q}_{i\alpha_{2l}}(t) = \sum_{j \in \mathcal{N}_i} a_{ij}(q_{j\alpha_{2l}}(t) - q_{i\alpha_{2l}}(t)) +$$

$$\sum_{j \in \mathcal{N}_i} a_{ij}([a_{be1i}(t)]_{lj}e_{1j}(t) - [a_{be1i}(t)]_{li}e_{1i}(t))$$

send  $e_{1i}(t)$ ,  $q_{ia_{b1}}(t)$ ,  $r_{ia_{e1}}(t)$ ,  $q_{ia_{e1}}(t)$ ,  $q_{i\alpha_{2l}}(t)$ ,  $l = 1, \dots, N$

---

B. DISTRIBUTED TRANSLATIONAL CONTROL

Now, consider the designed control law for the translational subsystem, that is  $U_v(t)$  in (30). The  $i$ -th element of  $U_v(t)$  is the altitude control law for  $i$ -th agent,  $u_{vi}(t)$ , which can be written as:

$$u_{vi}(t) = [B_4(t)]_{ii}^{-1}(\dot{\alpha}_{4i}(t) - \frac{e_{3i}(t)}{a_{b3}(t)} - \frac{1}{a_{e3}(t)} \sum_{j \in \mathcal{N}_i} (e_{3i}(t) - e_{3j}(t)) - k_4 e_{4i}(t)), \quad (38)$$

where  $A_4(t) = [\alpha_{41}^T(t), \dots, \alpha_{4N}^T(t)]^T$ . Similar to  $u_{1i}(t)$ , the main issue in computation of  $u_{vi}(t)$  is distributed calculation of  $a_{b3}(t)$ ,  $a_{e3}(t)$ , and  $\alpha_{4i}(t)$  as the  $i$ -th agent is only aware

of its own states  $e_{3i}(t)$  and  $e_{4i}(t)$  and the neighboring states  $e_{3j}(t), j \in \mathcal{N}_i$ .

1) DISTRIBUTED COMPUTATION OF  $a_{b3}(t)$  AND  $a_{e3}(t)$

Similar to  $a_{b1}(t)$ ,  $a_{b3}(t)$  can be calculated by computing  $E_3^T(t)E_3(t)$  using the consensus filter of (1). Toward this,  $r_i(t) = e_{3i}^T(t)e_{3i}(t)$  and the corresponding ones in neighboring agents,  $r_j(t) = e_{3j}^T(t)e_{3j}(t), j \in \mathcal{N}_i$ , are computed and are substituted in (1) with the consensus state of  $q_{ia_{b3}}(t)$ . The output of the consensus system,  $l_{ia_{b3}}(t)$  is then multiplied by  $N$  and is used to compute  $a_{b3i}(t) = k_{b3}^2 - Nl_{ia_{b3}}(t)$ . Moreover, in order to compute  $a_{e3}(t)$ , it is required to compute  $k_{e3}^2 - E_3^T(t)(\mathcal{L} \otimes I_2)E_3(t)$  in a distributed way. Similar to (36), consider the following equation:

$$\sum_{i,j=1}^N a_{ij} \|y_i - y_j\|^2 = 2y^T(\mathcal{L} \otimes I_2)y, \quad (39)$$

where  $y = [y_1^T, \dots, y_N^T]^T$  is a  $2N \times 1$  matrix where  $y_i = [y_{ix}, y_{iy}]^T$  is a  $2 \times 1$  matrix. Accordingly, each agent computes  $r_{ia_{e3}}(t) = \sum_{j \in \mathcal{N}_i} \|e_{3i}(t) - e_{3j}(t)\|^2$  and receives the corresponding ones  $r_{ja_{e3}}(t), j \in \mathcal{N}_i$  and uses (1) to compute  $\frac{2}{N}E_3^T(t)(\mathcal{L} \otimes I_2)E_3(t)$  with the consensus state of  $q_{ia_{e3}}(t)$ . Then, multiplying the consensus system output,  $l_{ia_{e3}}(t)$ , by  $\frac{N}{2}$ ,  $E_3^T(t)(\mathcal{L} \otimes I_2)E_3(t)$  is obtained and simply  $a_{e3i}(t) = k_{e3} - \frac{N}{2}l_{ia_{e3}}(t)$  is computed which asymptotically converges to  $a_{e3}(t)$ .

2) DISTRIBUTED COMPUTATION OF  $\alpha_{4i}(t)$

According to (31), to compute the  $2 \times 1$  vector of  $\alpha_{4i}(t)$ , it is required to compute  $(2i - 1)$ -th and  $(2i)$ -th elements of  $a_{be3}(t) = (a_{e3}(t)I_{2N} + a_{b3}(t)(\mathcal{L} \otimes I_2))^{-1}E_3(t)$ . Toward this, each agent separately computes  $a_{be3i}(t) = (a_{e3i}(t)I_{2N} + a_{b3i}(t)(\mathcal{L} \otimes I_2))^{-1}$  and replaces  $r_i(t) = a_{be3i}^{(2i-1:2i, 2i-1:2i)}(t)e_{3i}(t)$  and  $r_j(t) = a_{be3i}^{(2i-1:2i, 2j-1:2j)}(t)e_{3j}(t), i \in \mathcal{N}_i$  in (1) with consensus state of  $q_{ia_{4i}}(t)$ . Finally, the computed  $\frac{1}{N} \sum_{j=1}^N a_{be3i}^{(2i-1:2i, 2j-1:2j)}(t)e_{3j}(t)$  is clearly multiplied by  $N$  which gives  $\alpha_{4i}$  as follows:

$$\alpha_{4i}(t) = \dot{x}_{3id}(t) - Nk_3 a_{b3i}(t) a_{e3i}(t) l_{i\alpha_{4i}}(t). \quad (40)$$

The pseudo code of Table 1 can be easily extended for this procedure.

V. ATTITUDE CONTROL

In this section, we are going to solve **Problem 3** in which the proposed attitude control is presented for each agent. Toward this, using  $u_{vi}(t) = [u_{xi}(t), u_{yi}(t)]^T$  in (38), the desired trajectory of  $\phi_{id}(t)$  and  $\theta_{id}(t)$  can be generated as follows according to (11):

$$\begin{aligned} \sin(\phi_{id}(t)) &= u_{xi}(t) \sin(\psi_i(t)) - u_{yi}(t) \cos(\psi_i(t)), \\ \sin(\theta_{id}(t)) &= \frac{u_{xi}(t) \sin(\psi_i(t)) + u_{yi}(t) \sin(\psi_i(t))}{\cos(\phi_i(t))}, \end{aligned} \quad (41)$$

and  $\psi_{id}(t)$  can be freely chosen. It is worth mentioning that similar to [26], BLFs are employed to control the attitude

subsystem. However, the main and most important difference is that in [26], the equation (7), representing the relationship between angular velocities in the body and inertial frames has not been considered. Therefore, the system structure for the attitude control is more complicated and realistic in this paper. Toward this goal, (6) and (7) are firstly rewritten as follows:

$$\begin{aligned} \dot{x}_{5i}(t) &= g_i(x_{5i}(t))x_{6i}(t), \\ \dot{x}_{6i}(t) &= f_i(x_{6i}(t)) + B_{6i}u_i(t), \end{aligned} \quad (42)$$

where  $x_{5i}(t) = [\phi_i(t), \theta_i(t), \psi_i(t)]^T$ ,  $x_{6i}(t) = [p_i(t), q_i(t), r_i(t)]^T$ ,

$$\begin{aligned} g_i(x_{5i}(t)) &= \begin{bmatrix} 1 & \sin(\phi_i(t)) \tan(\theta_i(t)) & \cos(\phi_i(t)) \tan(\theta_i(t)) \\ 0 & \cos(\theta_i(t)) & -\sin(\phi_i(t)) \\ 0 & \frac{\sin(\phi_i(t))}{\cos \theta_i(t)} & \frac{\cos(\phi_i(t))}{\cos \theta_i(t)} \end{bmatrix}, \\ f_i(x_{6i}(t)) &= \begin{bmatrix} a_{1i}q_i(t)r_i(t) - a_{2i}\Omega_{ri}(t)q_i(t) \\ a_{3i}p_i(t)r_i(t) + a_{4i}\Omega_{ri}(t)p_i(t) \\ a_{5i}p_i(t)q_i(t) \end{bmatrix}, \end{aligned}$$

$B_{6i} = \text{diag}(b_{li}, l = 1, 2, 3)$ , and  $u_i(t) = [u_{2i}(t), u_{3i}(t), u_{4i}(t)]^T$ . As presented in Section II-B, the desired trajectory vector  $x_{5id}(t) = [\phi_{id}(t), \theta_{id}(t), \psi_{id}(t)]^T$  is firstly defined and the goal is  $\|e_{5i}(t)\| < k_{b5i}$  where  $e_{5i}(t) := x_{5i}(t) - x_{5id}(t)$  and  $k_{b5i}$  is a positive constant. Moreover,  $e_{6i}(t) := x_{6i}(t) - A_{6i}(t)$ . Now, the following BLF candidate is selected as presented in (5):

$$V_{5i}(e_{5i}(t)) = \frac{1}{2} \log \frac{k_{b5i}^2}{k_{b5i}^2 - e_{5i}^T(t)e_{5i}(t)}. \quad (43)$$

Therefore:

$$\dot{V}_{5i}(t) = \frac{e_{5i}^T(t) \left( g_i(x_{5i}(t))(e_{6i}(t) + A_{6i}(t)) - \dot{x}_{5id}(t) \right)}{a_{b5i}(t)}, \quad (44)$$

where  $a_{b5i}(t) = \|k_{b5i}\|^2 - e_{5i}^T(t)e_{5i}(t)$ . Thus,  $A_{6i}(t)$  is designed as follows:

$$A_{6i}(t) = \left( g_i(x_{5i}(t)) \right)^{-1} (\dot{x}_{5id}(t) - k_{5i} a_{b5i}(t) e_{5i}(t)), \quad (45)$$

where  $k_{5i}$  is a positive constant scalar. This selection gives:

$$\dot{V}_{5i}(t) = \frac{e_{5i}^T(t) g_i(x_{5i}(t)) e_{6i}(t)}{a_{b5i}(t)} - k_{5i} e_{5i}^T(t) e_{5i}(t), \quad (46)$$

where the first term should be canceled out in the next step. Now, a quadratic Lyapunov function is augmented to  $V_{5i}(e_{5i}(t))$  to cancel the additional term as follows:

$$V_{6i}(e_{5i}(t), e_{6i}(t)) = V_{5i}(e_{6i}(t)) + \frac{1}{2} e_{6i}^T(t) e_{6i}(t). \quad (47)$$

This selection gives:

$$\begin{aligned} \dot{V}_{6i}(t) &= \dot{V}_{5i}(t) + e_{6i}^T(t) \dot{e}_{6i}(t) \\ &= \dot{V}_{5i}(t) + e_{6i}^T(t) (f_i(x_{6i}(t)) + B_{6i}u_i(t) - \dot{A}_{6i}(t)). \end{aligned} \quad (48)$$

Therefore,  $u_i(t)$  is designed as follows:

$$\begin{aligned} u_i(t) &= B_{6i}^{-1} \left( -f_i(x_{6i}(t)) + \dot{A}_{6i}(t) - \frac{g_i(x_{5i}(t))e_{5i}(t)}{a_{b5i}(t)} \right. \\ &\quad \left. - k_{6i}e_{6i}(t) \right), \end{aligned} \quad (49)$$



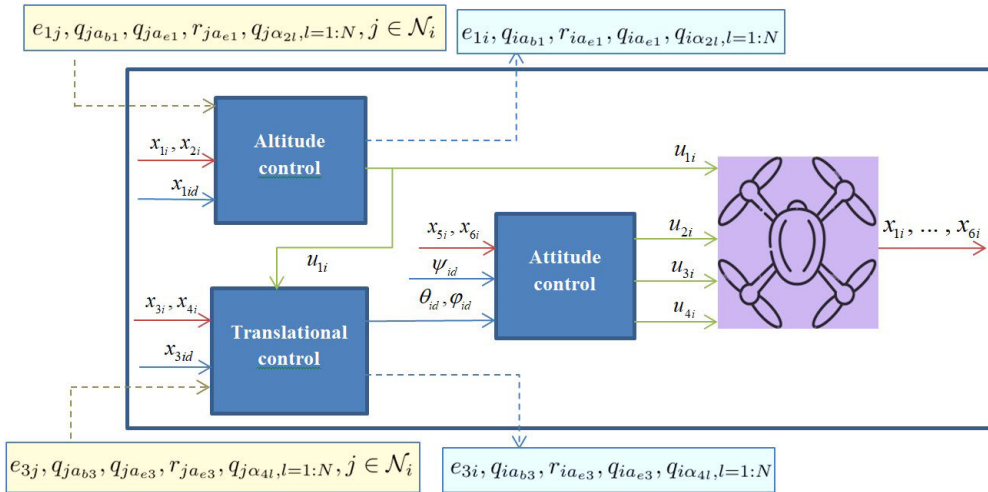


FIGURE 3. General framework of the proposed controller for  $i$ -th agent.



FIGURE 4. Interconnection between agents in the simulation.

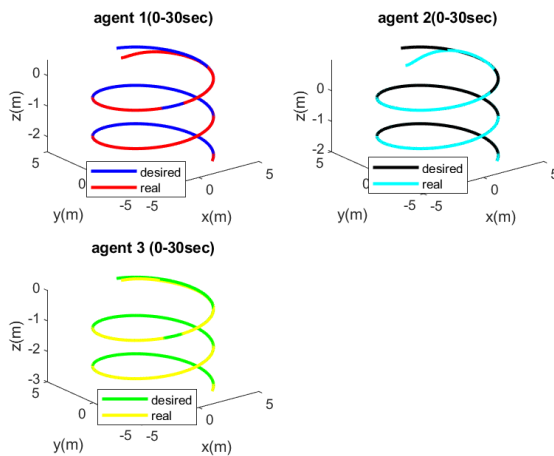


FIGURE 5. 3D controlled versus desired trajectories of the agents corresponding to the centralized case in the first scenario.

where  $k_{6i}$  is a positive constant scalar. Therefore,  $\dot{V}_{6i}(t) = -k_{5i}e_{5i}^T(t)e_{5i}(t) - k_{6i}e_{6i}^T(t)e_{6i}(t) < 0$  and it can be concluded that  $\|e_{5i}(t)\| < k_{b5i}, \forall t \geq 0$ , if  $\|e_{5i}(0)\| < k_{b5i}$ .

In order to clarify the proposed method, Figure 3 depicts the general framework of the proposed method for the  $i$ -th agent where dashed lines are related to data that should be sent to or received by the neighboring agents and solid lines are the internal signals where the blue ones are related to the desired signals, the green ones are the control signals and the red ones are the states measurements from the quadcopters.

## VI. SIMULATION RESULTS

In this section, the efficiency of the proposed controllers (both centralized and distributed) is demonstrated through

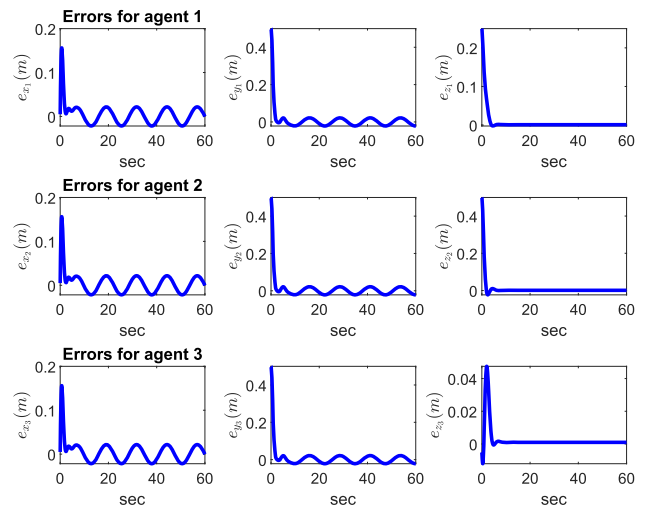


FIGURE 6. Position errors of the agents corresponding to the centralized case in the first scenario.

Matlab-Simulink simulation. Three interconnected quadcopters are considered where the interconnection of the agents are depicted in Figure 4.

The physical parameters of the quadcopters are assumed as  $I_{xxi} = 1.152 \times 10^{-2} \text{kgm}^2$ ,  $I_{yyi} = 1.152 \times 10^{-2} \text{kgm}^2$ ,  $I_{zz} = 2.18 \times 10^{-2} \text{kgm}^2$ ,  $I_{ri} = 5.2 \times 10^{-5} \text{kgm}^2$ ,  $l_i = 0.28 \text{m}$ ,  $m_i = 1.47 \text{kg}$ .

Two different scenarios are considered to evaluate the proposed method. In the first one, helical paths in which the agents form a line is considered in which the reference trajectories for quadcopters are formulated as follows:

$$\begin{aligned} x_{d1}(t) &= 4 \sin(0.5t), y_{d1}(t) = 4 \cos(0.5t), \\ z_{d1} &= -0.1t + 0.5, \psi_{d1} = 0.1 \text{rad}, \\ x_{d2}(t) &= 4 \sin(0.5t), y_{d2}(t) = 4 \cos(0.5t), \\ z_{d2} &= -0.1t + 1, \psi_{d2} = 0.1 \text{rad}, \\ x_{d3}(t) &= 4 \sin(0.5t), y_{d3}(t) = 4 \cos(0.5t), \\ z_{d3} &= -0.1t, \psi_{d3} = 0.1 \text{rad}. \end{aligned} \quad (50)$$

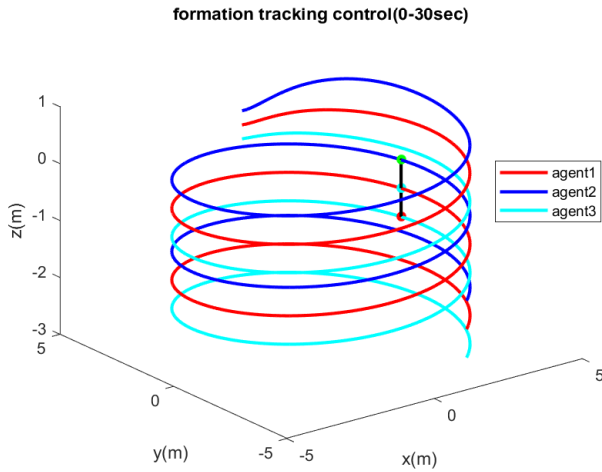


FIGURE 7. 3D formation tracking control and formation shape corresponding to the centralized case in the first scenario.

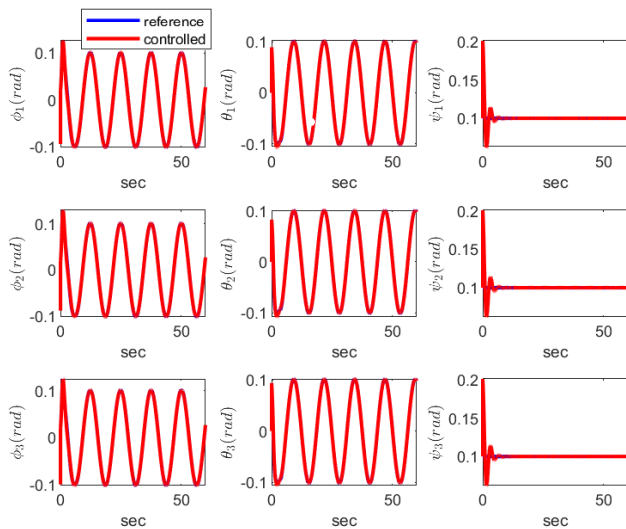


FIGURE 8. Controlled versus desired Euler angles corresponding to the centralized case in the first scenario.

In the second scenario, the quadcopters track the S-shape paths in a triangle formation shape. The reference trajectories are formulated in the following:

$$\begin{aligned}
 x_{d1}(t) &= 0.1t, y_{d1}(t) = 4 \cos(0.5t), \\
 z_{d1} &= 0.1t, \psi_{d1} = 0.1\text{rad}, \\
 x_{d2}(t) &= 0.1t + 0.5, y_{d2}(t) = 4 \cos(0.5t), \\
 z_{d2} &= 0.1t + 1, \psi_{d2} = 0.1\text{rad}, \\
 x_{d3}(t) &= 0.1t + 1, y_{d3}(t) = 4 \cos(0.5t), \\
 z_{d3} &= 0.1t, \psi_{d3} = 0.1\text{rad}.
 \end{aligned} \tag{51}$$

### A. FIRST SCENARIO

In this subsection, the first scenario is simulated and both of the centralized and decentralized approaches are evaluated from different aspects in details. Besides, comparison study with the method of [20] is provided. The initial conditions are considered as  $X_1(0) = [0.25, 0.5, 0.05]^T$ ,

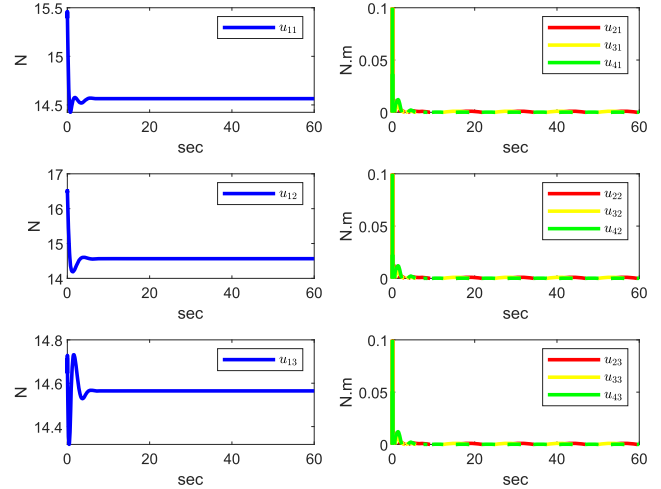


FIGURE 9. Control inputs corresponding to the centralized case in the first scenario.

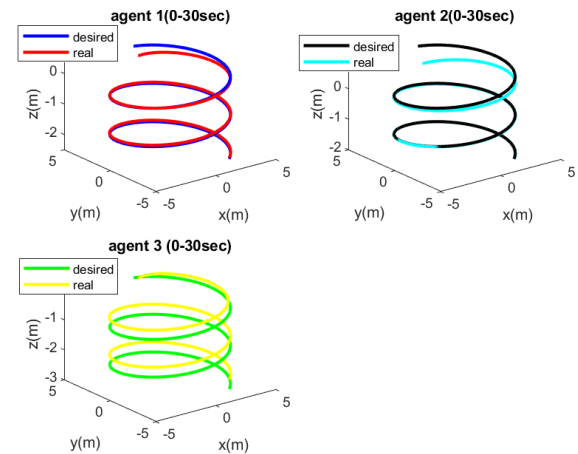


FIGURE 10. 3D controlled versus desired trajectories of the agents for 0 to 30 sec corresponding to the decentralized case in the first scenario.

$$\begin{aligned}
 X_2(0) &= [0.081, 0.081, 0.081]^T, X_3(0) = [0, 3.5, 0, 3.5, \\
 &0, 3.5]^T, X_4(0) = [1.5, 0, 1.5, 0, 1.5, 0]^T, x_{5i}(0) = [0, 0, \\
 &0.08]^T \text{ and } x_{6i}(0) = [0, 0, 0]^T, i = 1, 2, 3.
 \end{aligned}$$

### 1) CENTRALIZED FORMATION CONTROL

Firstly, the results for the centralized controller is presented. In this case, the parameters are selected as:  $k_{b1} = 1\text{m}$ ,  $k_{e1} = 0.43\text{m}$ ,  $k_{b3} = 1.12\text{m}$ ,  $k_{e3} = 0.14\text{m}$  and  $k_{b5} = 0.17\text{rad}$ . Besides,  $k_1 = k_2 = k_3 = k_4 = 1$  and  $k_{5i} = k_{6i} = 1$ ,  $i = 1, 2, 3$ .

Figure 5 depicts the 3D controlled (real) versus the desired trajectories of the agents. It is obvious from the figure that the method is successful to track the desired paths. Trajectory following errors tends to zero in a short time as depicted in Figure 6. In Figure 7 the result of formation tracking control and the formation shape of the agents are depicted. The controlled and desired Euler angles are also depicted in Figure 8 which also shows the accuracy of the method in controlling the Euler angles. Figure 9 shows the input signals. It is obvious that  $u_{1i}$ ,  $i = 1, 2, 3$  is less than 15N

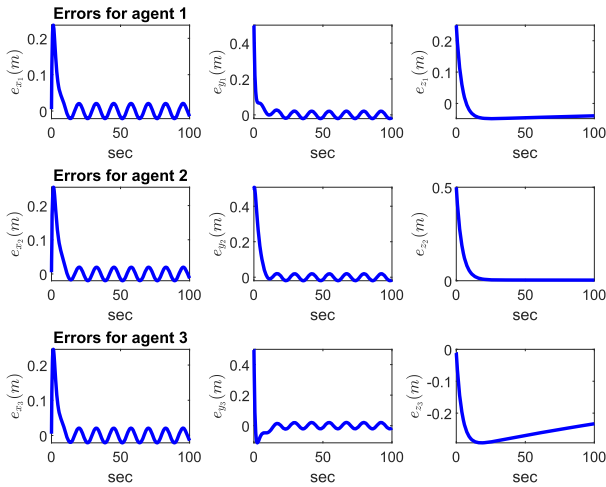


FIGURE 11. Position errors of the agents for 0 to 100 sec corresponding to the decentralized case in the first scenario.

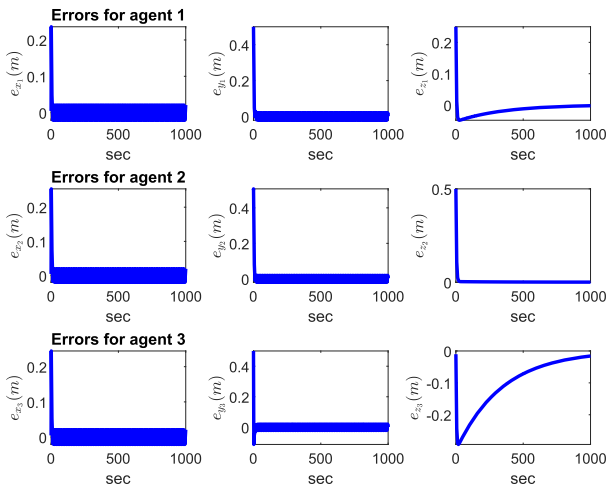


FIGURE 12. Position errors of the agents for 0 to 1000 sec corresponding to the decentralized case in the first scenario.

after transient time and  $u_{2i}, u_{3i}, u_{4i}, i = 1, 2, 3$  converges to 0N.m.

## 2) DECENTRALIZED FORMATION CONTROL

Now, the results for the proposed decentralized controller is presented. In this case, the parameters are selected as:  $k_{b1} = 10m, k_{e1} = 0.42m, k_{b3} = 12m, k_{e3} = 0.14m$  and  $k_{b5} = 0.17rad$ . Besides,  $k_1 = 10^{-5}, k_2 = 50, k_3 = 2.2 \times 10^{-5}, k_4 = 1$  and  $k_{5i} = k_{6i} = 10, i = 1, 2, 3$ .

Figure 10 depicts 3D controlled versus desired trajectories of the agents for the first 30sec. However, according to Figure 11, the  $z$  axis errors do not converges to zero. However, as it can be seen in Figure 12 the errors converges asymptotically to zero as time grows to 1000sec. It is due to the asymptotic behavior of the consensus algorithms. The behavior of agent 2 is different from two other ones as it is connected to two other agents at same time and this causes the errors converges to zero significantly faster.

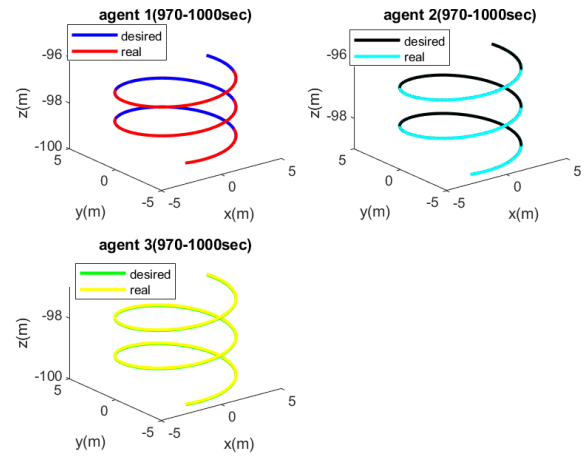


FIGURE 13. 3D controlled versus desired trajectories of the agent for 970 to 1000 sec corresponding to the decentralized case in the first scenario.

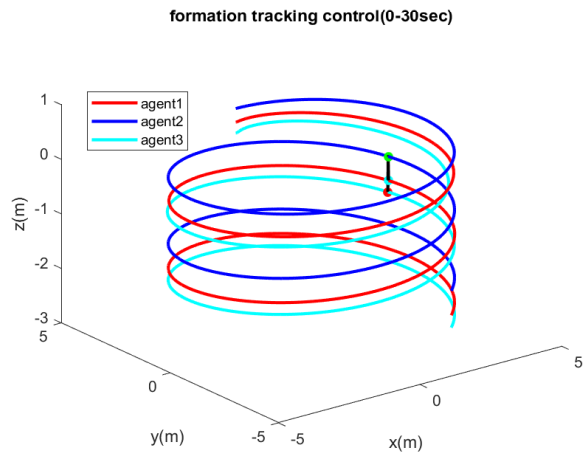


FIGURE 14. 3D formation tracking control and formation shape of the agents corresponding to the decentralized case in the first scenario.

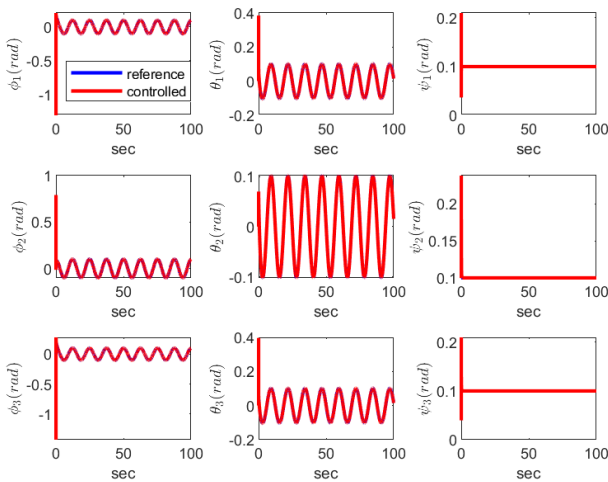
Figure 13 depicts the 3D controlled versus desired trajectories of the agents for the times between 970sec and 1000sec which shows a more precise tracking. Moreover, the result of formation tracking control of the agents for 30sec is depicted in Figure 14. The controlled and desired Euler angles are also depicted in Figure 15 for the first 100 sec which shows a precise tracking of the Euler angles. Figure 16 shows the input signals also for the first 100sec. It is obvious that like the centralized case in the decentralized one also  $u_{1i}, i = 1, 2, 3$  remains less than 15N after transient time and  $u_{2i}, u_{3i}, u_{4i}, i = 1, 2, 3$  converges to 0N.m. Figure 17 depicts distances between the agents. Although it has not been proved that the proposed controller is collision free, this can be investigated through simulation as depicted in Figure 17.

## 3) COMPARISON STUDY

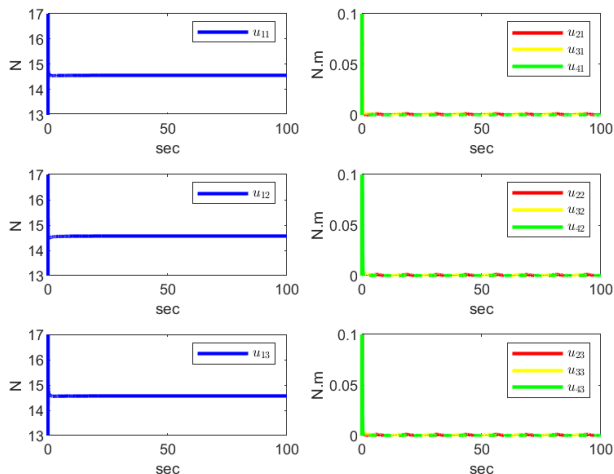
In this section, the proposed method of this paper is compared with the one presented in [20] where the finite-time formation

**TABLE 2.** The numerical comparison of our proposed method (centralized and decentralized ones) with the method of [20].

	RMSE (position)(m)	Torque (N.m)	Thrust force (N)	Steady state error (position) (m)	Settling time 5%(sec)
Our proposed method (centralized)	0.0273	0.0264	14.564	0.0218	1.92
Our proposed method (decentralized)	0.0357	0.0132	14.565	0.0206	385.4 (asymptotic convergence)
Method of [20]	0.0489	0.0463	14.558	0.0203	5.77

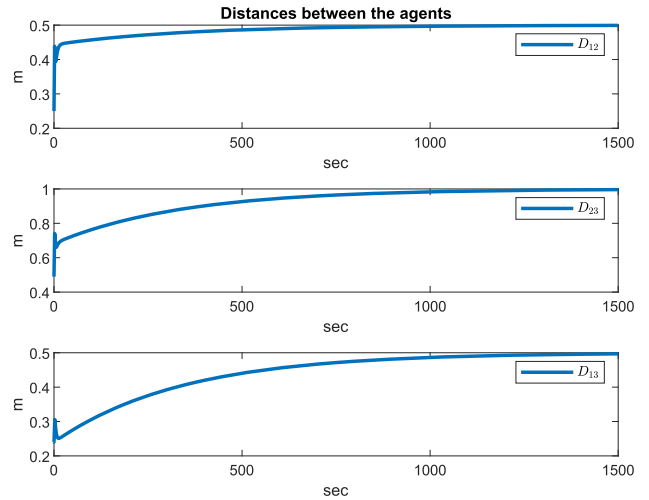


**FIGURE 15.** Controlled versus desired euler angles for 0 to 100 sec corresponding to the decentralized case in the first scenario.

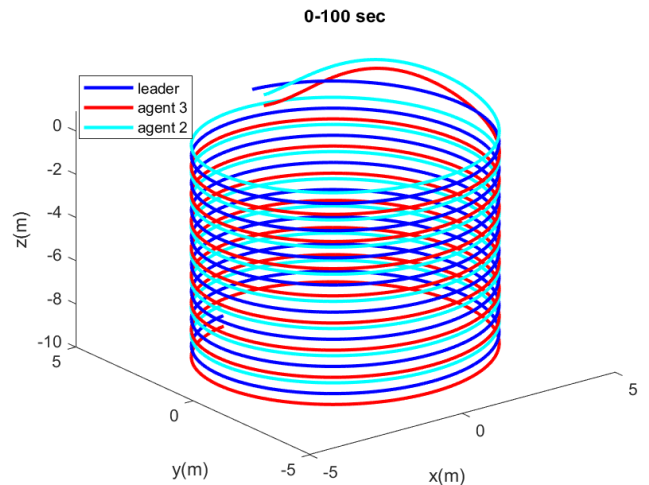


**FIGURE 16.** Control inputs for 0 to 100sec corresponding to the decentralized case in the first scenario.

tracking control is proposed in a leader-follower structure. In this paper, a special structure is considered for the topology



**FIGURE 17.** Distances between the agents corresponding to the decentralized case in the first scenario.



**FIGURE 18.** 3D controlled trajectories of the agents and the leader corresponding to the method of [20] in the first scenario.

of the agents and the leader in which it is assumed that the position of the leader should be in the average of the positions of the agents. Accordingly, we consider agent 1 as the leader to simulate the proposed method of [20]. Physical parameters (moments of inertia, weights and  $l_i$ ), initial values and desired trajectories are considered similar to our paper. Figure 18 depicts 3D controlled trajectories of agent 2 and agent 3 and the leader (agent 1 in our method). The errors are shown in Figure 19. It can be concluded from Figure 18 and Figure 19 that the method is successful in the formation tracking control in finite-time. However, comparing Figure 19 with Figure 11, it is obvious that although the convergence speed in the method of [20] is less than our proposed decentralized method (and not the centralized one), in its transient time, it suffers from severe under and overshoots which is not appropriate. Similarly, according to Figure 20, apart from its transient response the method of [20] is successful in tracking of the Euler angles. However, it can be seen from Figure 21

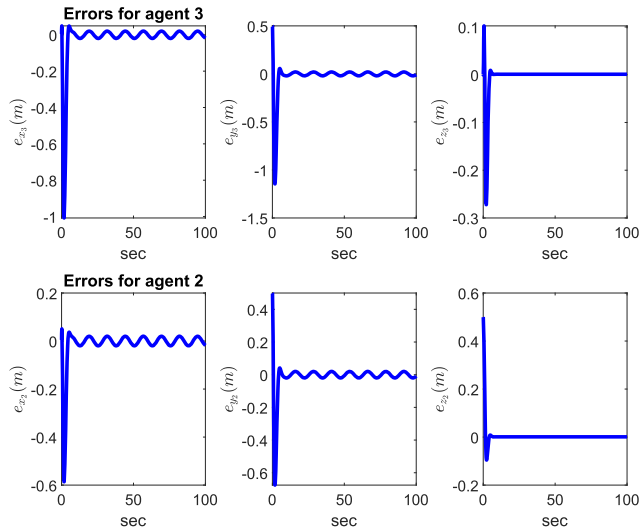


FIGURE 19. Position errors of the agents corresponding to the method of [20] in the first scenario.

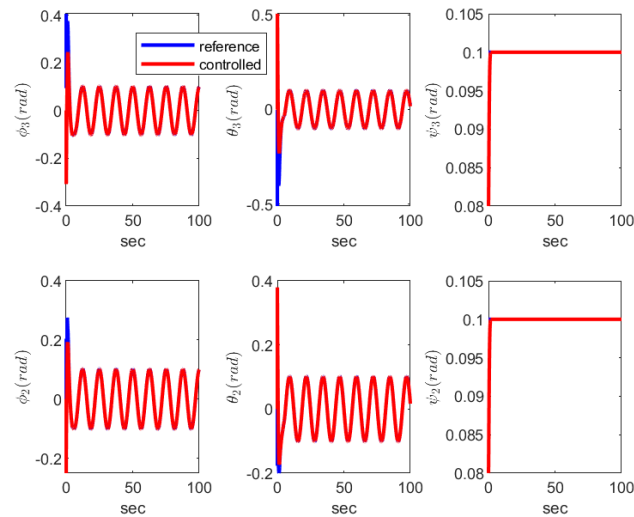


FIGURE 20. Controlled versus desired euler angles corresponding to the method of [20] in the first scenario.

that the higher level of torques are required to control the Euler angles rather than our proposed method (see Figure 16).

In order to provide a numerical comparison, Table 2 presents average of agents' root mean square errors (RMSE) and steady state errors in position, torques and thrust forces as well as the settling times for our proposed methods (centralized and decentralized) and the method of [20]. It is obvious from Table 2 that all of the methods are successful in the formation tracking control in the steady state according to their almost similar steady state errors. However, for our decentralized approach due the employment of asymptotically convergent consensus filters the settling time is significantly increased (as expected) while our proposed centralized method is significantly fast even almost 3 times faster the finite-time control method of [20]. The RMSE is the highest in the method of [20] due to large under and overshoots in its transient time and it is the

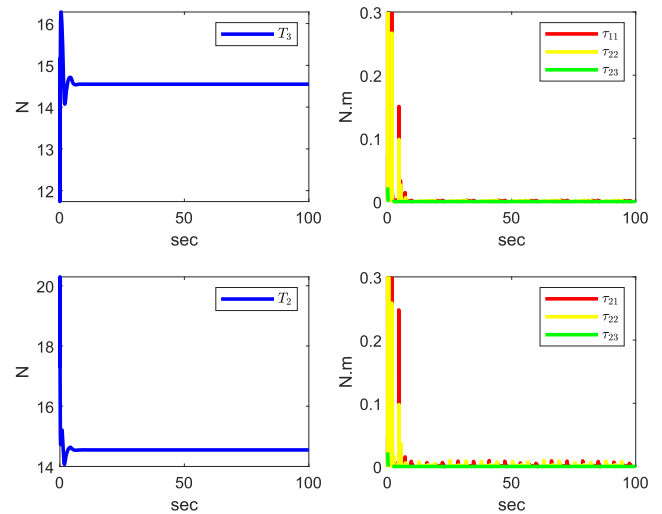


FIGURE 21. Control inputs corresponding to the method of [20] in the first scenario.

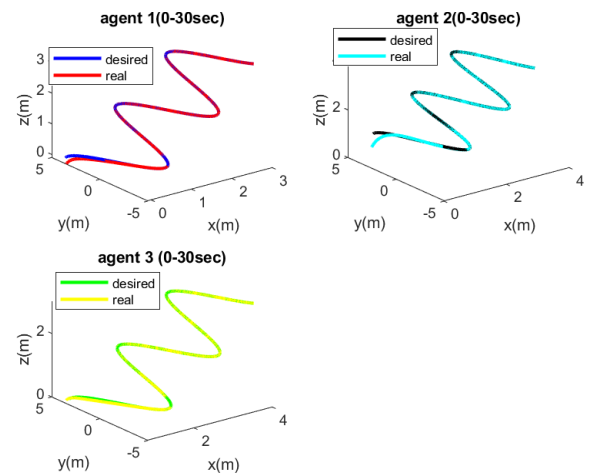


FIGURE 22. 3D controlled versus desired trajectories of the agents corresponding to the centralized case in the second scenario.

lowest in our centralized method. Although, the RMSE in our decentralized approach is smaller than the method of [20], it is bigger than the centralized one due to its slow convergence time. From the control effort perspective, our proposed decentralized method has better performance as it requires less torque for the attitude control.

### B. SECOND SCENARIO

In this subsection, the second scenario is simulated and both of the centralized and decentralized approaches are evaluated. The initial conditions are considered as  $X_1(0) = [-0.1, 0.6, 0.1]^T$ ,  $X_2(0) = [0.081, 0.081, 0.081]^T$ ,  $X_3(0) = [-0.1, 3.5, 0.3, 3.5, 0.8, 3.5]^T$ ,  $X_4(0) = [0.08, 0, 0.08, 0, 0.08, 0]^T$ ,  $x_{5i}(0) = [0, 0, 0.08]^T$  and  $x_{6i}(0) = [0, 0, 0]^T$ ,  $i = 1, 2, 3$ .

The 3D controlled versus the desired trajectories of the agents for both centralized and decentralized methods are depicted in Figure 22 and Figure 24, respectively, and the

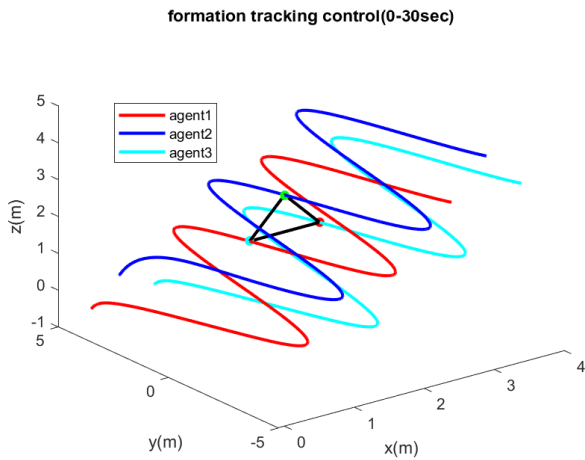


FIGURE 23. 3D formation tracking control and formation shape corresponding to the centralized case in the second scenario.

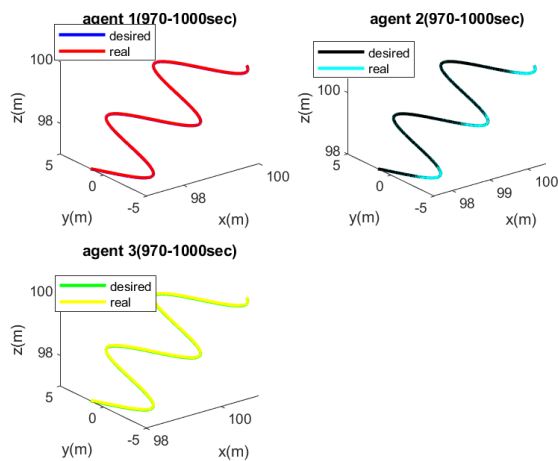


FIGURE 24. 3D controlled versus desired trajectories of the agents corresponding to the decentralized case in the second scenario.

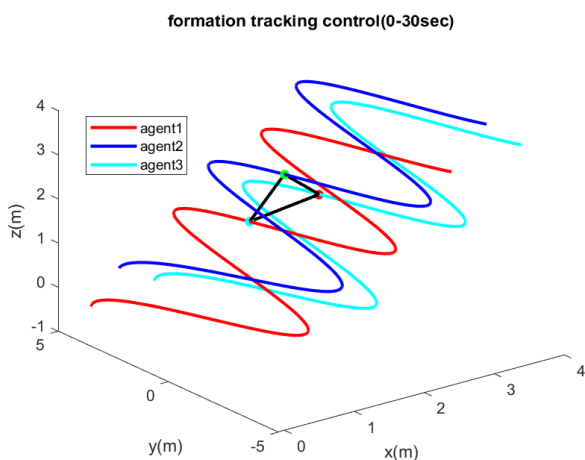


FIGURE 25. 3D formation tracking control and formation shape of the agents corresponding to the decentralized case in the second scenario.

3D formation tracking controls and formation shapes for both centralized and decentralized methods are depicted in Figure 23 and Figure 25, respectively.

## VII. CONCLUSION

The problem of formation tracking control in multi-quadcopter systems was studied in this paper. Toward this, firstly the problem was solved for a centralized case in which predefined accuracies in the trajectory tracking and formation control were satisfied through definition of logarithmic BLFs in a backstepping structure. Due to the under-actuated nature of the quadcopter system, at first, a centralized formation control was designed for the altitude subsystem and then through definition of virtual inputs, a centralized controller was designed translational subsystem. Then, the centralized controllers were implemented in a decentralized way using high pass consensus filters. Each agent, then employed the virtual inputs to generate the desired roll and pitch trajectories were the attitude control system to track the desired trajectories with a predefined accuracy were designed using logarithmic BLFs in a backstepping structure. Finally, the efficiency of the proposed method were demonstrated through simulation for both centralized and decentralized cases for two different spiral and S-shape scenario with line and triangle formation shpes, respectively. As expected, it was shown in the simulation that the results of the decentralized approach asymptotically converges to the centralized one due to the use of asymptotic convergent consensus filters. The proposed methods of this paper (both centralized and decentralized ones) were compared with the proposed method of [20] from different aspects of formation tracking precision both in transient and steady state times as well as control effort aspect.

Although it was shown in the simulation that the proposed method is collision free, the extension of the work to the collision free one with mathematical proofs is suggested as the future work. It is also proposed to extend the work to case in which the agents' controllers are derived in a distributed way at the beginning to omit the embedded consensus filters and make the method faster with less computational complexities. The proposed method can also be extended in several ways of multi-quadcopter systems with directional graph topology or for the systems in the presence of communication constraints such as protocol based or event triggered ones.

## REFERENCES

- [1] N. Sadeghzadeh-Nokhodberiz, A. Can, R. Stolkin, and A. Montazeri, "Dynamics-based modified fast simultaneous localization and mapping for unmanned aerial vehicles with joint inertial sensor bias and drift estimation," *IEEE Access*, vol. 9, pp. 120247–120260, 2021.
- [2] N. S. Nokhodberiz, H. Nemati, and A. Montazeri, "Event-triggered based state estimation for autonomous operation of an aerial robotic vehicle," *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 2348–2353, 2019.
- [3] N. Sadeghzadeh-Nokhodberiz, M. Iranshahi, and A. Montazeri, "Vision-based particle filtering for quad-copter attitude estimation using multirate delayed measurements," *Frontiers Robot. AI*, vol. 10, May 2023, Art. no. 1090174.
- [4] H. Do, H. Hua, M. Nguyen, C. Nguyen, H. Nguyen, H. Nguyen, and N. Nguyen, "Formation control algorithms for multiple-UAVs: A comprehensive survey," *EAI Endorsed Trans. Ind. Netw. Intell. Syst.*, vol. 8, no. 27, Jun. 2021, Art. no. 170230.
- [5] Y. Liu and R. Bucknall, "A survey of formation control and motion planning of multiple unmanned vehicles," *Robotica*, vol. 36, no. 7, pp. 1019–1047, Jul. 2018.

- [6] T. Xing, Y. Liu, and L. Chen, "Survey of robot formation control methods," *Academic J. Sci. Technol.*, vol. 4, no. 2, pp. 58–61, Jan. 2023.
- [7] V. Roldão, R. Cunha, D. Cabecinhas, C. Silvestre, and P. Oliveira, "A leader-following trajectory generator with application to quadrotor formation flight," *Robot. Auto. Syst.*, vol. 62, no. 10, pp. 1597–1609, Oct. 2014.
- [8] K.-H. Tan and M. A. Lewis, "Virtual structures for high-precision cooperative mobile robotic control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Mar. 1996, pp. 132–139.
- [9] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Trans. Robot. Autom.*, vol. 14, no. 6, pp. 926–939, Mar. 1998.
- [10] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.
- [11] N. Sadeghzadeh and A. Afshar, "New approaches for distributed sensor networks consensus in the presence of communication time delay," in *Proc. Chin. Control Decis. Conf.*, Jun. 2009, pp. 3632–3637.
- [12] N. Sadeghzadeh Nokhodberiz and J. Poshtan, "Belief consensus-based distributed particle filters for fault diagnosis of non-linear distributed systems," *Proc. Inst. Mech. Eng., I, J. Syst. Control Eng.*, vol. 228, no. 3, pp. 123–137, Mar. 2014.
- [13] W. Ren, "Consensus based formation control strategies for multi-vehicle systems," in *Proc. Amer. Control Conf.*, 2006, pp. 4237–4242.
- [14] B. Zhu, L. Xie, and D. Han, "Recent developments in control and optimization of swarm systems: A brief survey," in *Proc. 12th IEEE Int. Conf. Control Autom. (ICCA)*, Jun. 2016, pp. 19–24.
- [15] M. Zhang, X. Yang, Z. Xiang, and X. Liu, "Consensus of nonlinear MAS via double nonidentical mode-dependent event-triggered switching control," *Appl. Math. Comput.*, vol. 453, Sep. 2023, Art. no. 128085.
- [16] M. Wang, X. Yang, S. Mao, K. F. C. Yiu, and J. H. Park, "Consensus of multi-agent systems with one-sided Lipschitz nonlinearity via nonidentical double event-triggered control subject to deception attacks," *J. Franklin Inst.*, vol. 360, no. 9, pp. 6275–6295, Jun. 2023.
- [17] M. Zhang, X. Yang, Z. Xiang, and Y. Sun, "Monotone decreasing LKF method for secure consensus of second-order mass with delay and switching topology," *Syst. Control Lett.*, vol. 172, Feb. 2023, Art. no. 105436.
- [18] W. Ren and Y. Cao, *Distributed Coordination of Multi-Agent Networks: Emergent Problems, Models, and Issues*, vol. 1. Cham, Switzerland: Springer, 2011.
- [19] 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.
- [20] H. Du, W. Zhu, G. Wen, and D. Wu, "Finite-time formation control for a group of quadrotor aircraft," *Aerosp. Sci. Technol.*, vol. 69, pp. 609–616, Oct. 2017.
- [21] L. Dubois and S. Suzuki, "Formation control of multiple quadcopters using model predictive control," *Adv. Robot.*, vol. 32, no. 19, pp. 1037–1046, Oct. 2018.
- [22] N. Xuan-Mung and S. K. Hong, "Robust adaptive formation control of quadcopters based on a leader-follower approach," *Int. J. Adv. Robot. Syst.*, vol. 16, no. 4, 2019, Art. no. 1729881419862733.
- [23] X.-Z. Jin, W.-W. Che, Z.-G. Wu, and C. Deng, "Robust adaptive general formation control of a class of networked quadrotor aircraft," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 12, pp. 7714–7726, Dec. 2022.
- [24] K. P. Tee, S. S. Ge, and E. H. Tay, "Barrier Lyapunov functions for the control of output-constrained nonlinear systems," *Automatica*, vol. 45, no. 4, pp. 918–927, Apr. 2009.
- [25] X. Jin, S.-L. Dai, J. Liang, and D. Guo, "Multirobot system formation control with multiple performance and feasibility constraints," *IEEE Trans. Control Syst. Technol.*, vol. 30, no. 4, pp. 1766–1773, Jul. 2022.
- [26] S. Kumar and S. R. Kumar, "Barrier Lyapunov-based nonlinear trajectory following for unmanned aerial vehicles with constrained motion," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2022, pp. 1146–1155.
- [27] R. Dasgupta, S. B. Roy, O. S. Patil, and S. Bhasin, "A singularity-free hierarchical nonlinear quad-rotorcraft control using saturation and barrier Lyapunov function," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2019, pp. 3075–3080.
- [28] X. Li, H. Zhang, W. Fan, C. Wang, and P. Ma, "Finite-time control for quadrotor based on composite barrier Lyapunov function with system state constraints and actuator faults," *Aerosp. Sci. Technol.*, vol. 119, Dec. 2021, Art. no. 107063.
- [29] S. Ganguly, "Robust trajectory tracking and payload delivery of a quadrotor under multiple state constraints," 2022, *arXiv:2201.03711*.
- [30] M. Krstic, P. V. Kokotovic, and I. Kanellakopoulos, *Nonlinear and Adaptive Control Design*. Hoboken, NJ, USA: Wiley, Inc., 1995.
- [31] K. B. Ngo, R. Mahony, and Z.-P. Jiang, "Integrator backstepping using barrier functions for systems with multiple state constraints," in *Proc. 44th IEEE Conf. Decis. Control*, Jan. 2005, pp. 8306–8312.
- [32] R. Olfati-Saber and J. S. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," in *Proc. 44th IEEE Conf. Decis. Control*, Mar. 2005, pp. 6698–6703.
- [33] R. Olfati-Saber, "Distributed Kalman filter with embedded consensus filters," in *Proc. 44th IEEE Conf. Decis. Control*, Oct. 2005, pp. 8179–8184.
- [34] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *Proc. 46th IEEE Conf. Decis. Control*, Jan. 2007, pp. 5492–5498.
- [35] K. P. Tee, S. S. Ge, and F. E. H. Tay, "Adaptive control of electrostatic microactuators with bidirectional drive," *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 2, pp. 340–352, Mar. 2009.
- [36] T. Luukkonen, "Modelling and control of quadcopter," in *Independent Research Project in Applied Mathematics*, vol. 22. Espoo, Finland: Aalto Univ., 2011.



#### NARGESS SADEGHZADEH-NOKHODBERIZ

received the B.Sc. and M.Sc. degrees in control engineering from the Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran, in 2006 and 2008, respectively, and the Ph.D. degree in control engineering from the Iran University of Science and Technology, Tehran, in collaboration with the Automation Laboratory, Heidelberg University, Heidelberg, Germany, in September 2014. She was with the Department

of Engineering, Islamic Azad University Central Tehran Branch, Tehran, as an Assistant Professor, from 2015 to 2019. She was a Visiting Scholar with the Department of Electrical Engineering, Qatar University, Doha, Qatar, in 2016, and with the Department of Engineering, Lancaster University, Lancaster, U.K., in 2018. Since 2019, she has been an Assistant Professor with the Department of Electrical and Computer Engineering, Qom University of Technology, Qom, Iran. Her research interests include control systems theory, multi-agent systems, and mobile robot localization and mapping (SLAM).



#### NADER MESKIN (Senior Member, IEEE)

received the B.Sc. degree from the Sharif University of Technology, Tehran, Iran, in 1998, the M.Sc. degree from the University of Tehran, Tehran, in 2001, and the Ph.D. degree in electrical and computer engineering from Concordia University, Montreal, QC, Canada, in 2008. He was a Postdoctoral Fellow with Texas A&M University at Qatar, Doha, Qatar, from January 2010 to December 2010. He is currently an Associate

Professor with Qatar University, Doha, and an Adjunct Associate Professor with Concordia University. He has authored more than 190 refereed journals and conference papers. He is the coauthor (with K. Khorasani) of the book *Fault Detection and Isolation: MultiVehicle Unmanned Systems* (Springer, 2011). His research interests include fault detection and isolation, multi-agent systems, active control for clinical pharmacology, and linear parameter varying systems.

• • •