frontiers | Frontiers in Robotics and AI

TYPE Original Research
PUBLISHED 15 July 2024
DOI 10.3389/frobt.2024.1370104

Check for updates

Distributed safe formation tracking control of multiquadcopter systems using barrier Lyapunov function

Nargess Sadeghzadeh-Nokhodberiz[1]*, Mohammad Reza Sadeghi[1], Rohollah Barzamini[2] and Allahyar Montazeri[3]*

[1]Department of Control Engineering, Qom University of Technology, Qom, Iran, [2]Department of Electrical Engineering, Islamic Azad University Tehran Central Branch, Tehran, Iran, [3]School of Engineering, Lancaster University, Lancaster, United Kingdom

OPEN ACCESS

EDITED BY
Holger Voos,
University of Luxembourg, Luxembourg

REVIEWED BY
Önder Tutsoy,
Adana Science and Technology University, Türkiye
Xuejing Lan,
Guangzhou University, China

*CORRESPONDENCE
Nargess Sadeghzadeh-Nokhodberiz,
✉ sadeghzadeh@qut.ac.ir
Allahyar Montazeri,
✉ a.montazeri@lancaster.ac.uk

RECEIVED 13 January 2024
ACCEPTED 23 May 2024
PUBLISHED 15 July 2024

CITATION
Sadeghzadeh-Nokhodberiz N, Sadeghi MR, Barzamini R and Montazeri A (2024), Distributed safe formation tracking control of multiquadcopter systems using barrier Lyapunov function.
Front. Robot. AI 11:1370104.
doi: 10.3389/frobt.2024.1370104

Coordinating the movements of a robotic fleet using consensus-based techniques is an important problem in achieving the desired goal of a specific task. Although most available techniques developed for consensus-based control ignore the collision of robots in the transient phase, they are either computationally expensive or cannot be applied in environments with dynamic obstacles. Therefore, we propose a new distributed collision-free formation tracking control scheme for multiquadcopter systems by exploiting the properties of the barrier Lyapunov function (BLF). Accordingly, the problem is formulated in a backstepping setting, and a distributed control law that guarantees collision-free formation tracking of the quads is derived. In other words, the problems of both tracking and interagent collision avoidance with a predefined accuracy are formulated using the proposed BLF for position subsystems, and the controllers are designed through augmentation of a quadratic Lyapunov function. Owing to the underactuated nature of the quadcopter system, virtual control inputs are considered for the translational (*x* and *y* axes) subsystems that are then used to generate the desired values for the roll and pitch angles for the attitude control subsystem. This provides a hierarchical controller structure for each quadcopter. The attitude controller is designed for each quadcopter locally by taking into account a predetermined error limit by another BLF. Finally, simulation results from the MATLAB-Simulink environment are provided to show the accuracy of the proposed method. A numerical comparison with an optimization-based technique is also provided to prove the superiority of the proposed method in terms of the computational cost, steady-state error, and response time.

KEYWORDS

multiagent systems, formation control, intervehicle collision avoidance, barrier Lyapunov function (BLF), formation tracking control, backstepping controller

1 Introduction

Quadcopters are one of the most important categories of multirotor drones and consist of four arms, four motors, and four propellers. The control and navigation of quadcopters in a single or cooperative form have been the subject of various studies to enhance their capabilities for various applications (Montazeri et al., 2021; Sadeghzadeh-Nokhodberiz et al., 2023, 2021). A multiquadcopter system is a form of multiagent system that is used in various extreme environment applications, including nuclear decommissioning (Martin et al., 2016; Allahyar and Koubaa, 2023), volcanology (James et al., 2020), wildfire monitoring (Julian and Kochenderfer, 2019), and underground mining (Neumann et al., 2014). A multiagent system consists of several interacting intelligent agents that can cooperate their movements, sensing, and computations to achieve a common goal. Multiquadcopter systems are ideal solutions for different challenges imposed by humans working in extreme environments (Burrell et al., 2018); for example, using multiple quadcopters improves the performances, time and energy efficiencies, coverage areas, and redundancies of multiple robots performing the same task (Mansfield and Montazeri, 2024).

One of the most important issues in controlling a multiagent system is formation control to achieve consensus. Formation control is an important consideration in coordinating the control of a group of unmanned robots or quadcopters in the present study. It is assumed that each drone can fly and share information with the other robots in its neighborhood. Formation control is used in many applications relevant to environmental monitoring, such as coverage, patrolling, autonomous exploration, search and rescue, source seeking, and boundary tracking (Mansfield and Montazeri, 2024). In Liu and Bucknall (2018), the problem of formation control and cooperative motion planning of multiple unmanned vehicles was investigated and various approaches were reviewed; this work provides good insights into the challenges and techniques available for cooperative path planning and formation control.

One of the most investigated techniques to address the formation control problem is consensus-based formation control (Peng et al., 2020; Patil and Shah, 2021). Consensus is a displacement-based control mechanism, meaning that the agents simply need to know the relative locations (displacements) of their neighbors in a local reference system linked to a global system to achieve the desired formation. Displacement-based formation control is typically divided into three primary strategies: virtual structure (VS), behavior based (BB), and leader-follower (LF). The basic idea of consensus is that each vehicle updates its information state based on the information states of its local (time-varying) neighbors such that the final information state of each vehicle converges to a common value. The main purpose of a distributed formation control technique is to derive appropriate control commands for each agent based on the information provided by the agents that are only in the neighborhood of that agent. Here, the aim is that the team of robots should maintain a specific geometric shape while closely tracking the desired trajectory defined for the leader in the LF configuration or virtual leader in the formation control setting (Can et al., 2022; Imran and Montazeri, 2022). In such scenarios, the desired trajectory of each robot in a robotic fleet is not defined separately; instead, the trajectory should be defined, for example, for the center of the quad formation shape, under the connectivity assumption of the system graph that all agents can coordinate with the leader. Although this is a fully decentralized configuration, less centralized scenarios have also been reported in literature (Lizzio et al., 2022), in which the navigation was carried out at the ground control station and the desired trajectory was then transmitted to each drone that communicates with the neighboring agents to share their position and run the distributed on-board control algorithm to attain the desired trajectories. We adopted one such configuration in our investigation in this work.

The basic form of a formation control algorithm does not take into account the possibility of agent collisions while the agents attempt to reach their intended positions. For this reason, formation approaches considering interagent collision and/or obstacle avoidance have been the subject of investigations by some researchers. A comprehensive review comparing various collision avoidance strategies for unmanned aerial vehicle (UAV) applications can be found in Yasin et al. (2020). In the context of consensus-based formation control, similar collision avoidance strategies were surveyed and discussed by Sadeghzadeh-Nokhodberiz et al. (2023). When two drones generate a formation, they may collide with each other and obstacles in the transient phase as well as when reaching their desired positions and orientations. The consensus-based collision-free methods reviewed in both Lizzio et al. (2022) and Sadeghzadeh-Nokhodberiz et al. (2023) can be grouped under four main categories: (i) optimization-based techniques, (ii) force-field or artificial potential field (APF) techniques, (iii) geometric approaches, and (iv) sense-and-avoid approaches. As reviewed in Sadeghzadeh-Nokhodberiz et al. (2023), each of these approaches has its advantages/disadvantages for application to real life. Operationally speaking, static situations wherein obstacles are known ahead of time or are picked up on by the entire formation are more suited for optimization-based techniques. Although the force-field and geometric approaches are more effective for handling dynamic settings, the former may result in local minima due to cancellation of several APFs. Furthermore, the computational burden of a geometric technique in a busy dynamic environment may be very high when computing the collision-free trajectories. In terms of the operational requirements, optimization-based techniques are typically utilized in situations when the swarm must adhere to a predetermined reference trajectory. Depending on the algorithm chosen, each drone in the swarm either has a preloaded path or is aware of all the other reference trajectories. Nonetheless, the force-field and geometric techniques typically depend more on relative sensing or interagent communications.

Among the APF approaches reported for collision-free formation control, the method proposed by Yan et al. (2017) is notable because it causes the control signal to be limited and affected by the type of the potential field. Liang et al. (2020) studied a network of swarm drones, in which they followed a collision-free path by considering the system uncertainty in the presence of network constraints; the APF method was adopted here to address possible collisions between the UAVs, leading to a limited control signal. An example of an optimization-based technique used to design a collision-free formation control was reported by Kuriki and Namerikawa (2015); here, the problem was studied through the design of a consensus-based model predictive controller (MPC) by assuming that each UAV was located in a safe space and that the control input was updated as needed. The asymptotic stability of the proposed control method was also studied in detail. However, this method relied on the linear model of the system where the control system fails if the communication with the leader fails. Moreover, the collision avoidance strategy was considered only in the vertical direction. Jin et al. (2021) proposed a new framework to address the formation control of multiple robots; here, two types of problems were studied, namely the performance issues as well as feasibility of implementing the constraints when their requirements were in the tracking errors and distances between the paths.

Recently, reinforcement learning (RL) and deep reinforcement learning (DRL) techniques have been proven to be effective for decision-making and operation of cooperative robots in complex environments under time-varying and uncertain conditions. For example, Mansfield and Montazeri (2024) reviewed different multiagent RL (MARL) techniques used as advanced tools in the design of optimal cooperative trajectories for multirobot systems in environmental monitoring applications by optimizing not only the individual rewards of each of the robots but also their collective reward; although the focus of this was control and not desired trajectory design for quads operating in uncertain and complex environments, it was assumed that the target trajectory of each robot was designed and made available using the techniques of Liu and Bucknall (2018) or Mansfield and Montazeri (2024). Further, as mentioned in Mansfield and Montazeri (2024), the RL technique can be used to avoid interagent collisions and obstacles.

The above works do not use the barrier Lyapunov function (BLF) as an effective tool for collision-free formation tracking of quadcopters. More recently, Sadeghzadeh-Nokhodberiz and Meskin (2023) presented the problem of consensus-based formation tracking of multiquadcopter systems using logarithmic BLFs; however, the problem of collision avoidance was not considered. Instead, the method involved the use of a centralized approach that was then transformed to distributed control using highpass consensus filters. Although the performance of the proposed distributed method asymptotically converged to that of the centralized one, the convergence time was rather large. Therefore, the problem of collision-free formation tracking control of multiquadcopter systems is derived from scratch in a distributed manner in the present work.

Generally, the BLF is used to prevent the states from violating the constraints. Therefore, the BLF can be used to ensure safety and collision avoidance while guaranteeing convergence with a predefined accuracy. The BLF is a positive-definite function that grows to infinity when its arguments approach certain limits. Kumar and Kumar (2022) discussed the three-dimensional trajectory tracking problem of an unmanned vehicle with restrictions on the flight path during operations; to ensure that the quadrotor followed the desired trajectory while satisfying the imposed motion constraints, a BLF approach was proposed. Moreover, a six-degrees-of-freedom dynamic model of the system was considered to achieve high-accuracy tracking performance; this controller could avoid singularities in the attitude subsystem. Tang et al. (2013) proposed a single-input single-output non-linear control system using the BLF to avoid deviating from the safety range. Tee and Ge (2011) presented a feedback control system design with constraints on the states. Chen et al. (2020) studied the problem of obstacle avoidance for a system with multiple agents avoiding obstacles in the environment; in this method, a hybrid decentralized monitoring controller that guarantees collision avoidance was proposed. The method is scalable and can be applied to general non-linear robot dynamics. Recently, advanced model-based and uncertain optimal control laws have been developed and implemented in real time for impaired UAVs (Ahmadi et al., 2023). Ganguly (2022) used the BLF technique to design a controller for an N-degrees-of-freedom Euler–Lagrange system and numerically evaluated its effectiveness; this method was recently used for multirobot applications for interagent collision avoidance and tracking using second-order kinematics in two-dimensional cases (Jin et al., 2021). It is worth mentioning that Khadhraoui et al. (2023) and Mughees and Ahmad (2023) used BLFs for single quadcopter systems, in addition to Sadeghzadeh-Nokhodberiz and Meskin (2023), who recently employed BLFs for formation tracking of multiquadcopter systems without considering the collision avoidance problem.

Based on the above literature, a decentralized (distributed) collision-free formation tracking control method is proposed in this work for cooperative control of quadcopter systems. The proposed method is used for interagent collision avoidance and trajectory tracking with a predefined accuracy. Compared to the aforementioned works, the proposed method has a lower computational burden, is easily scalable, and can be used in dynamic environments. Further, contrary to the APF approaches, the proposed method does not limit the control signal for collision avoidance. The major contribution of the present study is that the barrier Lyapunov method is used to derive a distributed collision-free formation tracking control in which both formation tracking and interagent collision avoidance are considered simultaneously. Accordingly, BLFs are first proposed for the position subsystems (x, y, and z axes) and controllers are designed by augmenting a quadratic Lyapunov function, leading to a backstepping procedure. Owing to the underactuated nature of the quadcopter system, virtual inputs are considered for the translational (x and y axes) subsystems that are then used to generate the desired values for the roll and pitch angles for the attitude control subsystem. This provides a hierarchical controller structure for each quadcopter.

The distributed formation tracking controller derived herein not only guarantees convergence of the formation tracking error with a predefined accuracy but also avoids interagent collisions during the transient responses of the formation. Thus, both collision avoidance and trajectory tracking with a predefined bound on the tracking error are achieved in a distributed manner. The novelty of this work is briefly summarized as follows:

- Formulating multiple problems, including trajectory tracking, formation tracking control, and interagent collision avoidance, of a multiquadcopter system using the proposed BLF.
- Deriving decentralized (distributed) hierarchical control laws for collision-free formation tracking control of the altitude as well as translational x and y axes subsystems using virtual inputs in a backstepping framework.
- Designing attitude control laws separately for each agent using desired signals generated via BLFs while considering a predefined accuracy.

The remainder of this article is organized as follows. Section 2 details the problem formulation and preliminaries. Section 3 presents a decentralized collision-free formation tracking controller design for a multiquadcopter system using BLFs. Section 4 presents the simulation results, and Section 5 contains a summary of the conclusions.

2 Preliminaries and problem formulation

This section presents some preliminaries on the required theoretical materials.

2.1 Graph theory

Consider the graph $G = \{V, \xi, W\}$ containing N nodes, where $V = \{1, 2, \ldots, N\}$ is the set of nodes and ξ is the set of all the edges of the graph. It is assumed that the edge (i,j) between nodes i and j exists, where i and j are adjacent to each other, such that $\xi = (i,j) \in V \times V$. If $(i,j) \in \xi \Leftrightarrow (j,i) \in \xi$, then the graph is undirected. Matrix $A = \left[a_{ij}\right]$ is the adjacency matrix such that if there is a path from i to j in the system graph, then $a_{ij} = a_{ji} = 1$. A path from i to j is a sequence of distinct nodes starting at i and ending at j, such that each pair of consecutive nodes is adjacent. If there is a path from i to j, then the nodes are connected. If all the paths of a graph are connected, then the graph is connected. The degree matrix of a graph D is a diagonal matrix with elements d_i that are equal to the set of neighboring nodes. $N_i = \{j \in V{:}(i,j) \in \xi\}$, where N_i is the set of neighbors surrounding i. The matrix L is the Laplacian matrix of the graph that is equal to $L = D - A$, and the sum of it rows is equal to zero (Hu et al., 2021).

2.2 Barrier Lyapunov theory

Consider the non-linear system given by Eq. (1) as follows:

$$\begin{aligned} \dot{x}_1(t) &= f_1(x_1(t)) + g_1(x_1(t))x_2(t), \\ \dot{x}_2(t) &= f_2(x_1(t), x_2(t)) + g_1(x_1(t), x_2(t))u(t), \end{aligned} \tag{1}$$

where $x_1(t) \in \mathbb{R}^{n_1}$ and $x_2(t) \in \mathbb{R}^{n_2}$ are system states, $u(t) \in \mathbb{R}^{n_2}$ is the system input, and vector functions f_1, f_2, g_1, and g_2 are assumed to be smooth. The goal here is to design the control law $u(t)$ such that $x_1(t)$ follows the desired trajectory $x_{1d}(t)$ with a predefined accuracy. In other words, if $e_1(t) := x_1(t) - x_{1d}(t)$, the control objective is to ensure that the tracking error remains within a compact set defined by $D_1 = \{e_1(t) \in \mathbb{R}^{n_1} | d_1(t) = \|e_1(t)\| < \Omega_1, t \geq 0\}$, where Ω_1 is a predefined positive scalar. Next, the idea of using the BLF in a backstepping procedure (Tee et al., 2009; Ngo et al., 2005; Tee et al., 2008) is extended to the vector form case. Therefore, assuming that the BLF $V_1(t)$ is defined as in Eq. (2), the Lyapunov candidate function $V(t)$ is defined by augmenting $V_2(t)$ to $V_1(t)$ as follows:

$$\begin{aligned} V_1(t) &= \frac{1}{2}\eta_1^2(t), \\ V(t) &= V_1(t) + V_2(t), \end{aligned} \tag{2}$$

where $\eta_1(t) = \frac{\Omega_1 d_1(t)}{\Omega_1 - d_1(t)}$ and $V_2 = \frac{1}{2}{z_2}^T z_2$ with $z_2(t) = x_2(t) - \alpha(t)$ is defined as an auxiliary tracking error for the virtual control input; $\alpha(t)$ is a stabilizing vector function that must be designed. According to Tee et al. (2009) and Lemma 1 therein, if the inequality $\dot{V}(t) \leq 0$ holds $\forall t \geq 0$, it can be concluded that $e_1(t) \in D_1$ if $e_1(0) \in D_1$.

2.3 Quadcopter model

Assume we have a group of quadcopters consisting of N agents communicating with each other. The dynamic of the attitude subsystem of the ith quadcopter (assuming a small Euler angle) for $i = 1, \ldots, N$ can be written as follows:

$$\begin{aligned} \ddot{\phi}_i(t) &= \frac{I_{yy_i} - I_{zz_i}}{I_{xx_i}}\dot{\theta}_i(t)\dot{\psi}_i(t) - I_{r_i}\Omega_{r_i}\frac{\dot{\theta}_i(t)}{I_{xx_i}} + \frac{u_{2i}(t)}{I_{xx_i}}, \\ \ddot{\theta}_i(t) &= \frac{I_{zz_i} - I_{xx_i}}{I_{yy_i}}\dot{\phi}_i(t)\dot{\psi}_i(t) + I_{r_i}\Omega_{r_i}\frac{\dot{\phi}_i(t)}{I_{yy_i}} + \frac{u_{3i}(t)}{I_{yy_i}}, \\ \ddot{\psi}_i(t) &= \frac{I_{zz_i} - I_{xxi}}{I_{yy_i}}\dot{\phi}_i(t)\dot{\theta}_i(t) + \frac{u_{4i}(t)}{I_{zz_i}}. \end{aligned} \tag{3}$$

where the roll angle $\phi_i(t)$, pitch angle $\theta_i(t)$, and yaw angle $\psi_i(t)$ represent the rotations about the x, y, and z axes in the inertial frame, respectively. The input signals $u_{2i}(t)$, $u_{3i}(t)$, and $u_{4i}(t)$ represent torques in the corresponding directions for the ith quadcopter in the body frame. I_{xx_i}, I_{yy_i}, and I_{zz_i} are the inertia tensors, and I_{r_i} is the inertia of the propellers. Further, Ω_{r_i} describes the relative speed of the propeller.

The translational dynamics of the ith quadcopter can be presented as follows:

$$\begin{aligned} \ddot{x}_i(t) &= \frac{u_{1i}(t)}{m_i}\left(\cos\left(\psi_i(t)\right)\sin\left(\theta_i(t)\right)\cos\left(\phi_i(t)\right) + \sin\left(\psi_i(t)\right)\sin\left(\phi_i(t)\right)\right), \\ \ddot{y}_i(t) &= \frac{u_{1i}(t)}{m_i}\left(\sin\left(\psi_i(t)\right)\sin\left(\theta_i(t)\right)\cos\left(\phi_i(t)\right) - \cos\left(\psi_i(t)\right)\sin\left(\phi_i(t)\right)\right), \\ \ddot{z}_i(t) &= -g + \frac{u_{1i}(t)}{m_i}\cos\left(\phi_i(t)\right)\cos\left(\theta_i(t)\right), \end{aligned} \tag{4}$$

where $\left[x_i(t) \quad y_i(t) \quad z_i(t)\right]^T$ represents the position of ith quadcopter in the inertial frame, $u_{1i}(t)$ defines the main thrust created by the combined forces of the rotors, g is the gravitational constant, and m_i refers to the mass of the ith quadcopter (Sadeghzadeh-Nokhodberiz et al., 2021).

The above dynamic system can be represented in the state-space form, and the system is divided into three subsystems for simplicity as altitude, translational, and attitude subsystems (Sadeghzadeh-Nokhodberiz et al., 2021).

The altitude subsystem can be decomposed as follows:

$$\begin{aligned} \dot{x}_{1i}(t) &= x_{2i}(t), \\ \dot{x}_{2i}(t) &= -g + g_{2i}(t)u_{1i}(t), \end{aligned} \tag{5}$$

where $x_{1i}(t) \equiv z_i(t)$ and $x_{2i}(t) \equiv \dot{z}_i(t)$ refer to the altitude and velocity of the ith quadcopter in the z direction, respectively; $u_{1i}(t)$ is the control input indicating the thrust force applied to the ith quadcopter in the z direction; $g_{2i}(t) = \frac{1}{m_i}\cos\left(\theta_i(t)\right)\cos\left(\phi_i(t)\right)$ is an auxiliary variable defined to convert the last expression in Eq. (4) to a more compact form.

The translational subsystem is defined as follows:

$$\begin{aligned} \dot{x}_{3i}(t) &= x_{4i}(t), \\ \dot{x}_{4i}(t) &= g_{4i}(t)u_{iv3}(t), \\ \dot{x}_{5i}(t) &= x_{6i}(t), \\ \dot{x}_{6i}(t) &= g_{6i}(t)u_{iv5}(t), \end{aligned} \tag{6}$$

where $x_{3i}(t) \equiv x_i(t)$ and $x_{4i}(t) \equiv \dot{x}_i(t)$ refer to the position and velocity of the ith quadcopter in the x direction, and $x_{5i}(t) \equiv y_i(t)$ and $x_{6i}(t) \equiv \dot{y}_i(t)$ refer to the position and velocity of the ith quadcopter in the y direction, respectively; $g_{4i}(t) = g_{6i}(t) = \frac{u_{1i}(t)}{m_i}$ are auxiliary variables defined to convert the last expression in Eq. (4) to a more compact form. Moreover, $u_{iv3}(t)$ and $u_{iv5}(t)$ are virtual controller inputs to enable control of the underactuated position subsystem and are defined as follows:

$$u_{iv3}(t) = \cos(\psi_i(t))\sin(\theta_i(t))\cos(\phi_i(t)) + \sin(\psi_i(t))\cos(\phi_i(t)) \quad (7)$$

$$u_{iv5}(t) = \sin(\psi_i(t))\sin(\theta_i(t))\cos(\phi_i(t)) - \cos(\psi_i(t))\sin(\phi_i(t)) \quad (8)$$

Finally, the attitude subsystem can be defined using Eq. (3) by assuming that I_{r_i} is very small:

$$\begin{aligned} \dot{x}_{7i}(t) &= x_{8i}(t), \\ \dot{x}_{8i}(t) &= f_{2i}(t) + G_{8i}u_i(t), \end{aligned} \quad (9)$$

where $x_{7i}(t) \equiv [\phi_i(t) \quad \theta_i(t) \quad \psi_i(t)]^T$ and $x_{8i}(t) \equiv [\dot{\phi}_i(t) \quad \dot{\theta}_i(t) \quad \dot{\psi}_i(t)]^T$ are the respective attitude and angular velocity vectors in the inertial frame; $u_i(t) = [u_{2i}(t) \quad u_{3i}(t) \quad u_{4i}(t)]^T$ is the control input vector including the torques in the corresponding directions for the ith quadcopter in the body frame; $f_{2i}(t) = [a_{1i}\dot{\theta}_i(t)\dot{\psi}_i(t) \quad a_{3i}\dot{\phi}_i(t)\dot{\psi}_i(t) \quad a_{5i}\dot{\phi}_i(t)\dot{\theta}_i(t)]^T$ is an auxiliary vector with auxiliary variables defined by $a_{1i} = \frac{I_{yy_i} - I_{zz_i}}{I_{xx_i}}$, $a_{3i} = \frac{I_{zz_i} - I_{xx_i}}{I_{yy_i}}$, and $a_{5i} = \frac{I_{zz_i} - I_{xxi}}{I_{yy_i}}$; $G_{8i} = \begin{bmatrix} b_{1i} & 0 & 0 \\ 0 & b_{3i} & 0 \\ 0 & 0 & b_{5i} \end{bmatrix}$ is an auxiliary matrix with auxiliary variables $b_{1i} = \frac{1}{I_{xx_i}}$, $b_{3i} = \frac{1}{I_{yy_i}}$, and $b_{5i} = \frac{1}{I_{zz_i}}$ defined to ensure that the attitude dynamics defined in Eq. (3) are in a compact form.

Now, the following problems are considered in this work:

Problem 1: Formulating multiple problems, including trajectory tracking, formation tracking control, and interagent collision avoidance, for a multiquadcopter system such that the proposed barrier Lyapunov theory can be applied.

Problem 2: Deriving the decentralized (distributed) hierarchical control laws for collision-free formation tracking control for the altitude subsystem as well as translational x and y subsystems with virtual inputs in a backstepping framework.

Problem 3: Designing the attitude control laws separately for each agent using the desired signals generated via BLFs while considering a predefined accuracy.

3 Control objectives

Problem 1 is considered in this section. In this work, the goal is to design controllers $u_{1i}(t), \ldots, u_{4i}(t)$ such that the control objectives are achieved. The control objective in this study is formation tracking control, which consists of two parts. First, each quadcopter should follow its specified desired trajectory with a predefined accuracy for position and orientation. Second, interagent collisions should be avoided based on specified bounds regarding how close the quadcopters can be.

3.1 Trajectory tracking error

Let $x_{1id}(t), i = 1, 2, \ldots, N$ (where N indicates the number of quadcopters) be the desired altitude trajectory for the ith quadcopter that is continuous in time and has finite first- and second-order derivatives. Further, we define the altitude tracking error for the ith quadcopter as $e_{1i}(t) = x_{1i}(t) - x_{1id}(t)$. The first control objective here is to ensure that the altitude of the ith quadcopter tracks the desired trajectory $x_{1id}(t)$ with a predefined accuracy; this can be formulated by ensuring that the altitude tracking error for the ith quadcopter remains with a compact set defined as follows:

$$D_{1ie} = \{e_{1i}(t) \in \mathbb{R} | d_{1ie}(t) = |e_{1i}(t)| < \Omega_{1idH}, t \geq 0\}, \quad (10)$$

where Ω_{1idH} is a positive scalar defined for the ith quadcopter with an upper bound for the tracking error.

Similar to the altitude, $x_{3id}(t)$ and $x_{5id}(t)$ are the desired translational trajectories for the ith quadcopter in the x and y directions, respectively. It is assumed that these desired trajectories are continuous in time and have limited first- and second-order derivatives. Further, $e_{3i}(t) = x_{3i}(t) - x_{3id}(t)$ and $e_{5i}(t) = x_{5i}(t) - x_{5id}(t)$ represent the tracking errors in the x and y directions for the ith quadcopter, respectively. The control objective here is to track the desired translational trajectories $x_{3id}(t)$ and $x_{5id}(t)$ with a predetermined accuracy; this can be formulated by ensuring that the translational tracking error for the ith quadcopter remains within a compact set defined as follows:

$$\begin{aligned} D_{3ie} &= \{e_{3i}(t) \in \mathbb{R} | d_{3ie}(t) = |e_{3i}(t)| < \Omega_{3idH}, t \geq 0\}, \\ D_{5ie} &= \{e_{5i}(t) \in \mathbb{R} | d_{5ie}(t) = |e_{5i}(t)| < \Omega_{5idH}, t \geq 0\}, \end{aligned} \quad (11)$$

where Ω_{3idH} and Ω_{5idH} are two separate positive scalars defined for the ith quadcopter in the x and y directions, respectively, with upper bounds for the tracking errors.

Finally, for the attitude subsystem, $x_{7id}(t)$ is considered as the desired trajectory vector for the ith quadcopter and assumed to be continuous in time with limited first- and second-order derivatives. Further, $e_{7i}(t) = x_{7i}(t) - x_{7id}(t)$ represents the attitude tracking error vector. The control objective here is to track the desired trajectory vector $x_{7id}(t)$ with a predefined accuracy; this can be formulated by ensuring that the attitude tracking error for the ith quadcopter remains within the compact set defined as follows:

$$D_{7ie} = \{e_{7i}(t) \in \mathbb{R}^3 | d_{7ie}(t) = \|e_{7i}(t)\| < \Omega_{7idH}, t \geq 0\}, \quad (12)$$

where Ω_{7idH} is a predefined positive scalar with an upper bound for the tracking error, and $\|.\|$ is the 2-norm of the vector.

3.2 Collision avoidance and formation control

As introduced earlier, $x_{1i}(t)$ is the altitude of the ith quadcopter with $x_{1j}(t), j \in N_i$ being the altitudes of its neighboring agents. The goal here is that the distances of the real altitudes of each of the agents with their neighbors, i.e., $d_{1ij}(t) \triangleq |x_{1i}(t) - x_{1j}(t)|, j \in N_i$, will track the desired distances, i.e., $L_{1ij}(t) \triangleq |x_{1id}(t) - x_{1jd}(t)|, j \in N_i$, which are expressed by $d'_{1ije}(t) \triangleq |d_{1ij}(t) - L_{1ij}(t)|, j \in N_i$ with a predefined accuracy. Thus, formation control and interagent collision avoidance

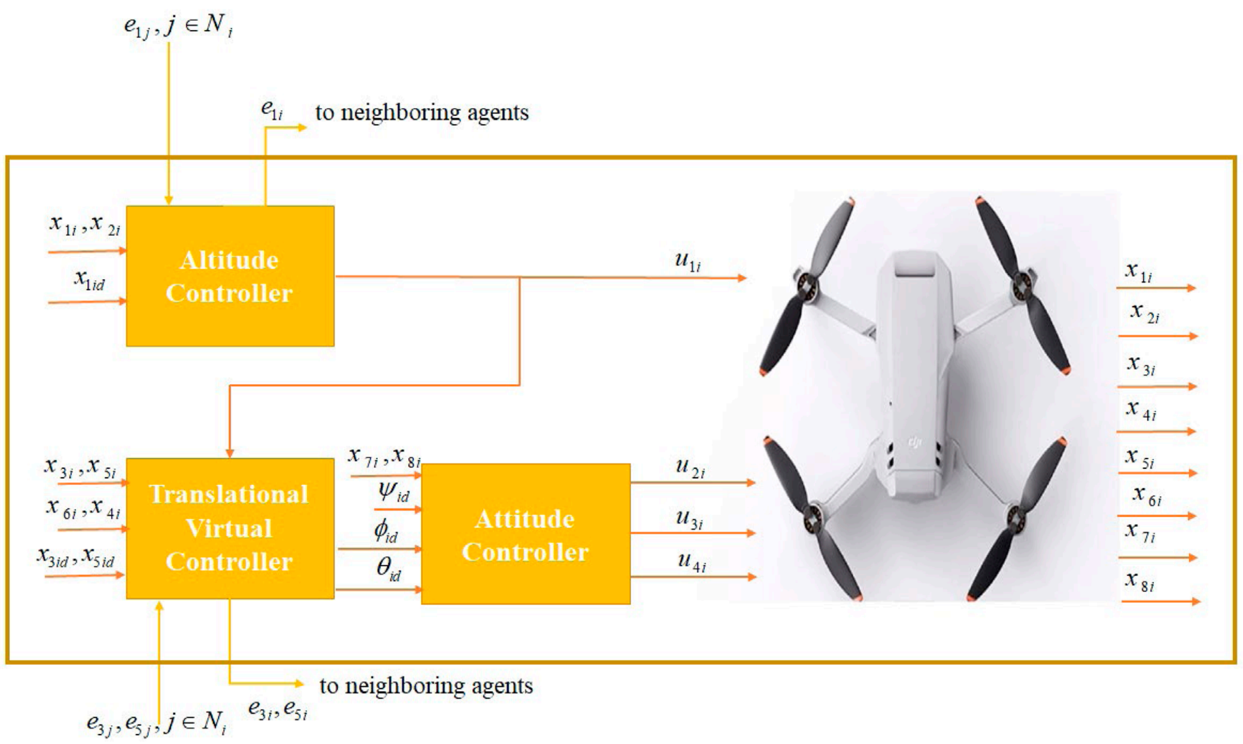

FIGURE 1
General framework of the proposed controller for the *i*th agent.

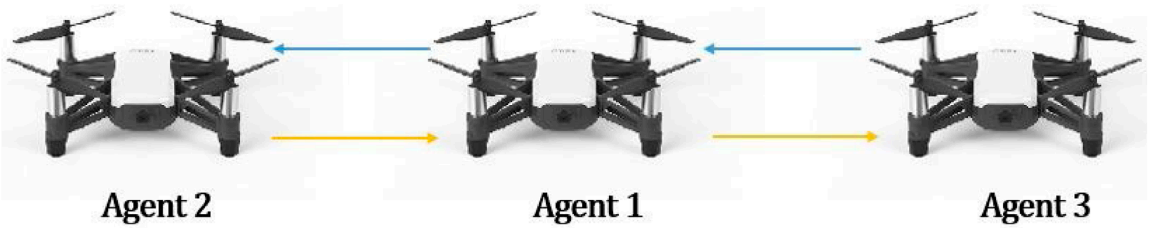

FIGURE 2
Interconnections between the agents in the simulation.

in the z direction are both guaranteed. This is achieved by ensuring that the error $d'_{1ije}(t)$ for the ith quadcopter remains within the compact set defined as follows:

$$D_{1ijH} = \left\{ x_{1i}(t) \in \mathbb{R} \middle| d'_{1ije}(t) < \Omega_{1ijH}, j \in N_i, t \geq 0 \right\}, \tag{13}$$

where Ω_{1ijH} is a positive predefined scalar with an upper bound for formation tracking and a collision avoidance bound.

The real distance of the ith quadcopter from its neighboring agents in the x direction is given by $d_{3ij}(t) \triangleq |x_{3i}(t) - x_{3j}(t)|, j \in N_i$, while the desired distance is represented by $L_{3ij}(t) \triangleq |x_{3id}(t) - x_{3jd}(t)|, j \in N_i$. Then, the goals of formation control and interagent collision avoidance in the x direction for the ith quadcopter are guaranteed with a predefined accuracy if $d'_{3ije}(t) \triangleq |d_{3ij}(t) - L_{3ij}(t)|, j \in N_i$ remains within the compact set defined as follows:

$$D_{3ijH} = \left\{ x_{3i}(t) \in \mathbb{R} \middle| d'_{3ije}(t) < \Omega_{3ijH}, j \in N_i, t \geq 0 \right\}, \tag{14}$$

where Ω_{3ijH} is a positive predefined scalar with an upper bound for formation tracking and a collision avoidance bound.

Similarly, the real distance of the ith quadcopter from its neighboring agents in the y direction is given by $d_{5ij}(t) \triangleq |x_{5i}(t) - x_{5j}(t)|, j \in N_i$, while the desired distance is represented by $L_{5ij}(t) \triangleq |x_{5id}(t) - x_{5jd}(t)|, j \in N_i$. Then, the goals of formation control and interagent collision avoidance in the y direction for ith quadcopter are guaranteed with a predefined accuracy if $d'_{5ije}(t) \triangleq |d_{5ij}(t) - L_{5ij}(t)|, j \in N_i$ remains within the compact set defined as follows:

$$D_{5ijH} = \left\{ x_{5i}(t) \in \mathbb{R} \middle| d'_{5ije}(t) < \Omega_{5ijH}, j \in N_i, t \geq 0 \right\}, \tag{15}$$

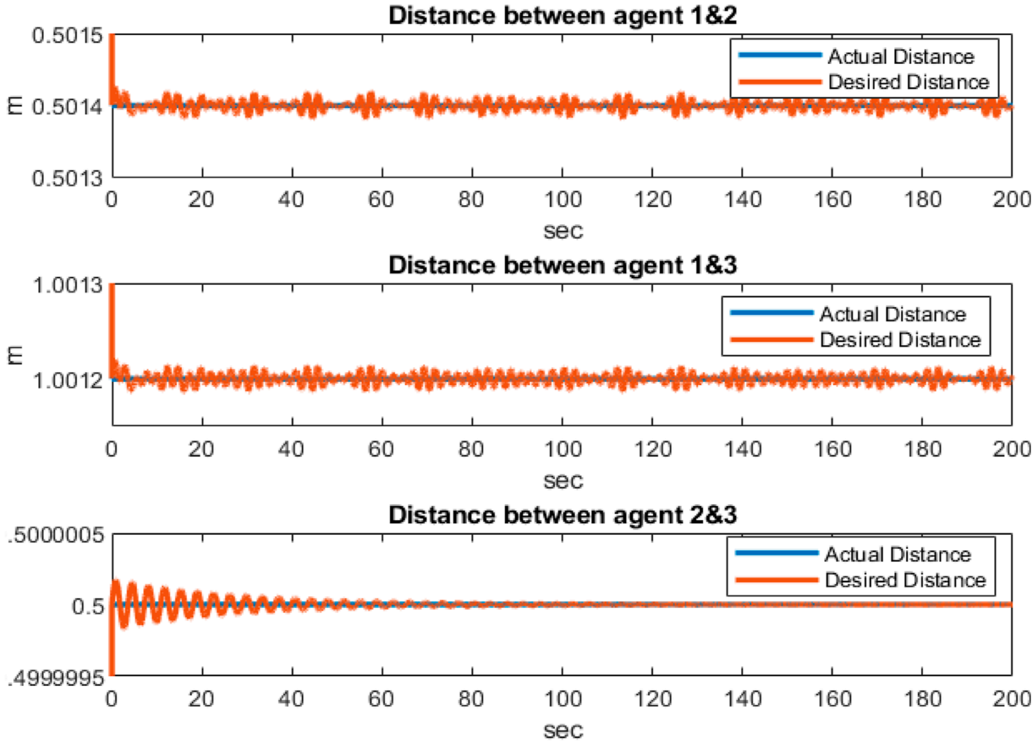

FIGURE 3
Distances between the agents in the simulation.

where Ω_{5ijH} is a positive predefined scalar with an upper bound for formation tracking and a collision avoidance bound.

Remark 1: It is worth noting that the problem considered here is neither LF control nor formation producing with/without a virtual leader. However, the shape of the formation is imposed on the method by appropriate design of the desired trajectory for each quadcopter.

4 Proposed distributed collision-free formation tracking control

Problem 2 is considered in this section, and the decentralized (distributed) hierarchical control laws for collision-free formation tracking control for the altitude and translational x and y subsystems with virtual inputs are designed in a backstepping framework for the multiquadcopter system. As mentioned earlier, because of the underactuated nature of the quadcopter system, a hierarchical procedure is employed. As the first step, the altitude controller is designed, and its result is used to design the controller for the translational subsystems along with virtual control inputs.

4.1 Altitude subsystem

Theorem 1: Assume that the altitude subsystem of the ith quadcopter in a fleet of N quadcopters is described by Eq. (5). Then, the altitude control input for the ith quadcopter $u_{1i}(t)$ can be designed as

$$u_{1i}(t) = g_{2i}^{-1}(t)\left[g + \dot{\alpha}_{1i}(t) - A_{1i}(t) - k_{2i} z_{2i}(t)\right]; \tag{16}$$

where

$$\alpha_{1i}(t) = \dot{x}_{1id}(t) + \frac{-k_{1i} e_{1i}^2(t)}{D_{1i}(t) e_{1i}(t) - 2\sum_{j\in N_i} B_{1ij}(t) e_{1j}(t)},$$

$$A_{1i}(t) = D_{1i}(t) e_{1i}(t) - 2\sum_{j\in N_i} B_{1ij}(t) e_{1j}(t),$$

$$D_{1i}(t) = B_{1i}(t) + 2\sum_{j\in N_i} B_{1ij}(t), \tag{17}$$

with $B_{1i} = \frac{\Omega_{1idH}^3}{(\Omega_{1idH} - d_{1ie}(t))^3}$, $B_{1ij} = \frac{\Omega_{1ijH}^3}{(\Omega_{1ijH} - d'_{1ije}(t))^3}$, $\Omega_{1ijH} > 0, \Omega_{1idH} > 0$, and $k_{1i}, k_{2i} > 0$. Then, the altitude tracking error and interagent collision avoidance conditions are guaranteed by remaining within the sets defined by Eqs. (10) and (13) if the quadcopter starts with the initial conditions such that the tracking errors remain within the same sets, i.e., $d_{1ie}(0) < \Omega_{1idH}$ and $d'_{1ije}(0) < \Omega_{1ijH}$, respectively.

Proof: We choose the following BLF candidate that contains the BLFs for each of the agents ($V_{1ie}(t)$) as well as those related to the interagents ($V_{1ij}(t)$):

$$V_{1i}(t) = \sum_{i=1}^{N}\left(V_{1ie}(t) + \sum_{j=1, j\neq i}^{N} a_{ij} V_{1ij}(t)\right), \tag{18}$$

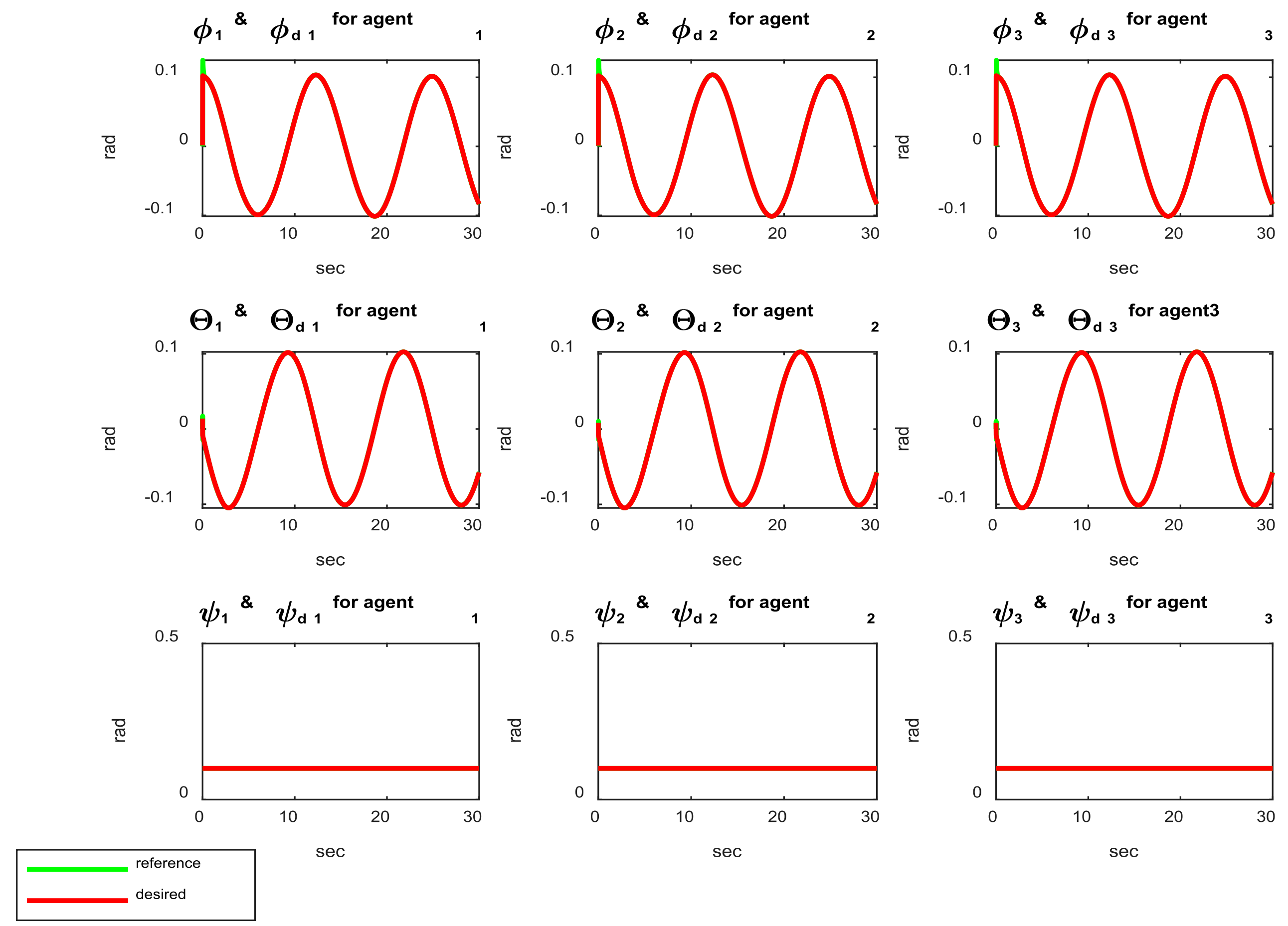

FIGURE 4
Comparison of the reference and actual trajectory states for the attitude subsystem.

where $V_{1ie}(t) = \frac{1}{2}\eta_{1ie}^2(t)$ and $V_{1ij}(t) = \frac{1}{2}\eta_{1ij}^2(t)$, with $\eta_{1ie}(t) = \frac{\Omega_{1idH}d_{1ie}(t)}{\Omega_{1idH}-d_{1ie}(t)}$ and $\eta_{1ij}(t) = \frac{\Omega_{1ijH}d'_{1ije}(t)}{\Omega_{1ijH}-d'_{1ije}(t)}$ according to Eq. (2).

It is obvious from Eq. (18) that V_{1i} is a positive-definite function. Therefore,

$$\begin{aligned}\dot{V}_{1i}(t) &= \sum_{i=1}^{N}\left(\dot{V}_{1ie}(t) + \sum_{j=1,j\neq i}^{N} a_{ij}\dot{V}_{1ij}(t)\right)\\ &= \sum_{i=1}^{N}\left(\eta_{1ie}(t)\dot{\eta}_{1ie}(t) + \sum_{j=1,j\neq i}^{N} a_{ij}\eta_{1ij}(t)\dot{\eta}_{1ij}(t)\right),\end{aligned} \tag{19}$$

where $\dot{V}_{1ie}(t) = \eta_{1ie}(t)\dot{\eta}_{1ie}(t) = \frac{\Omega_{1idH}^3}{(\Omega_{1idH}-d_{1ie}(t))^3}d_{1ie}(t)\dot{d}_{1ie}(t)$; further, by letting $B_{1i}(t) = \frac{\Omega_{1idH}^3}{(\Omega_{1idH}-d_{1ie}(t))^3}$, it can be concluded that $\dot{V}_{1ie}(t) = B_{1i}(t)d_{1ie}(t)\dot{d}_{1ie}(t)$. Since $d_{1ie} = |e_{1i}|$, we have

$$\begin{aligned}\dot{V}_{1ie}(t) &= B_{1i}(t)|e_{1i}(t)|\frac{d}{dt}|e_{1i}(t)|\\ &= B_{1i}(t)|e_{1i}(t)|\operatorname{sgn}\left(e_{1i}(t)\right)\dot{e}_{1i}(t)\\ &= B_{1i}(t)e_{1i}(t)\dot{e}_{1i}(t).\end{aligned} \tag{20}$$

Moreover, $\dot{V}_{1ij}(t) = \eta_{1ij}(t)\dot{\eta}_{1ij}(t) = \frac{\Omega_{1ijH}^3}{\left(\Omega_{1ijH}-d'_{1ije}(t)\right)^3}d'_{1ije}(t)\dot{d}'_{1ije}(t)$. Now, by letting $B_{1ij}(t) = \frac{\Omega_{1ijH}^3}{\left(\Omega_{1ijH}-d'_{1ije}(t)\right)^3}$, it is concluded that

$$\begin{aligned}\dot{V}_{ij}(t) &= B_{1ij}(t)d'_{ije}(t)\dot{d}'_{ije}(t)\\ &= B_{1ij}(t)\left(e_{1i}(t)-e_{1j}(t)\right)\left(\dot{e}_{1i}(t)-\dot{e}_{1j}(t)\right).\end{aligned} \tag{21}$$

Finally, by replacing Eqs. (20) and (21) in Eq. (19), we obtain

$$\dot{V}_{1i}(t) = \sum_{i=1}^{N}\left(B_{1i}(t)e_{1i}(t)\dot{e}_{1i}(t) + \sum_{j=1,j\neq i}^{N} a_{ij}B_{1ij}(t)\left(e_{1i}(t)-e_{1j}(t)\right)\left(\dot{e}_{1i}(t)-\dot{e}_{1j}(t)\right)\right). \tag{22}$$

By rearranging Eq. (22), we have

$$\begin{aligned}\dot{V}_{1i}(t) &= \sum_{i=1}^{N}B_{1i}(t)e_{1i}(t)\dot{e}_{1i}(t) + \sum_{i=1}^{N}\sum_{j=1}^{N}a_{ij}B_{1ij}(t)\left(e_{1i}(t)-e_{1j}(t)\right)\dot{e}_{1i}(t)\\ &\quad -\sum_{i=1}^{N}\sum_{j=1}^{N}a_{ij}B_{1ij}(t)\left(e_{1i}(t)-e_{1j}(t)\right)\dot{e}_{1j}(t).\end{aligned} \tag{23}$$

Using the summation properties and the fact that $a_{ij} = a_{ji}$ and $B_{1ij}(t) = B_{1ji}(t)$, it is concluded that $-\sum_{i=1}^{N}\sum_{j=1}^{N}a_{ij}B_{1ij}(t)\left(e_{1i}(t)-e_{1j}(t)\right)\dot{e}_{1j}$

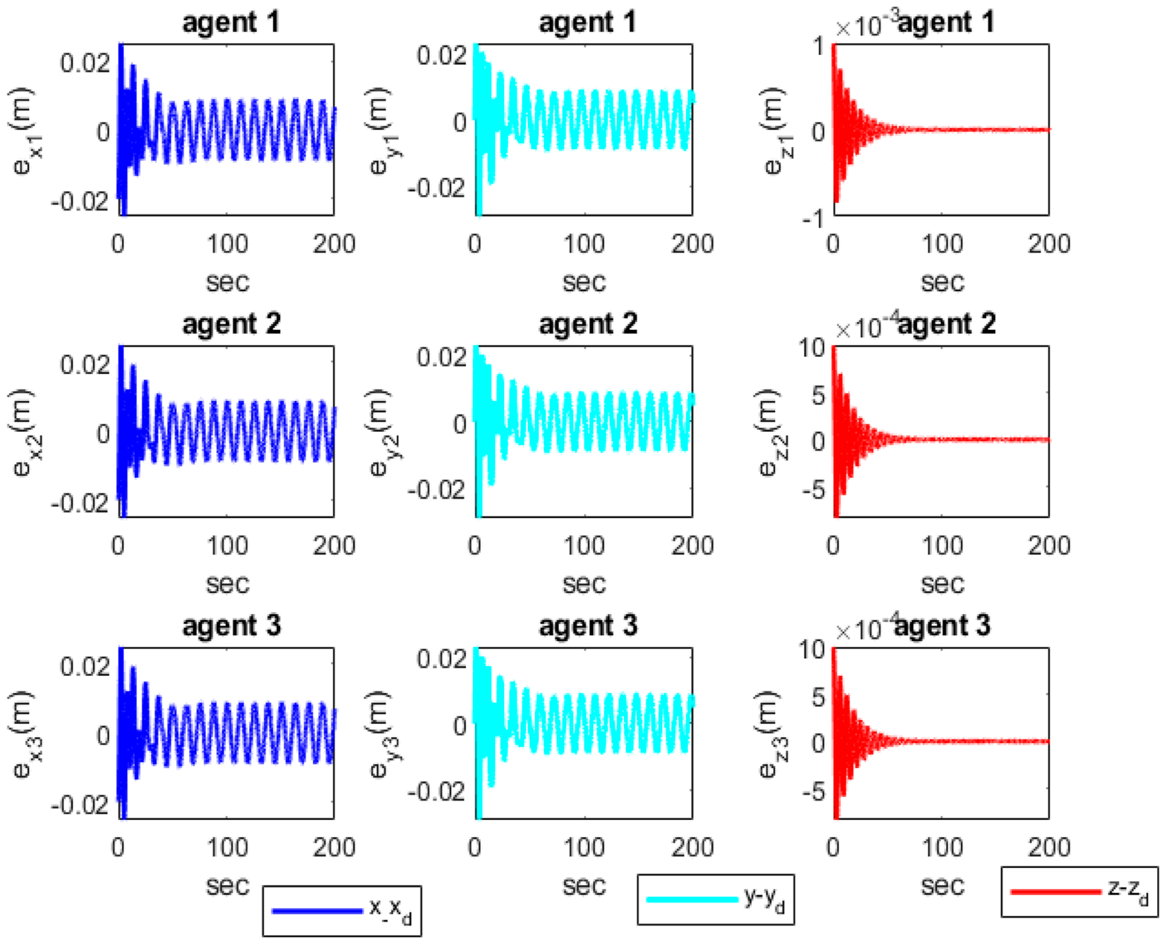

FIGURE 5
Position tracking errors of the agents in the *x*, *y*, and *z* axes.

$(t) = \sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij}B_{1ij}(t)\big(e_{1i}(t) - e_{1j}(t)\big)\dot{e}_{1i}(t)$; hence, Eq. (23) can be rewritten as

$$\dot{V}_{1i}(t) = \sum_{i=1}^{N}\left(B_{1i}(t)e_{1i}(t) + 2\sum_{j=1}^{N} a_{ij}B_{1ij}(t)\big(e_{1i}(t) - e_{1j}(t)\big)\right)\dot{e}_{1i}(t). \quad (24)$$

Assuming that $z_{2i}(t) = x_{2i}(t) - \alpha_{1i}(t)$, we have $x_{2i}(t) = z_{2i}(t) + \alpha_{1i}(t)$. Since $\dot{e}_{1i}(t) = x_{2i}(t) - \dot{x}_{1id}(t)$, the expression can be rewritten as $\dot{e}_{1i}(t) = z_{2i}(t) + \alpha_{1i}(t) - \dot{x}_{1id}(t)$. Now, substituting this into Eq. (24), the stabilizing function α_{1i} is derived as in Eq. (17). Therefore, Eq. (24) can be rewritten as $\dot{V}_{1i}(t) = \sum_{i=1}^{N}[A_{1i}(t)z_{2i} - k_{1i}e_{1i}^2(t)]$. By defining a backstepping-type Lyapunov function candidate and adding a quadratic function to $V_{1i}(t)$, we have $V_{2i}(t) = V_{1i}(t) + \frac{1}{2}\sum_{i=1}^{N} z_{2i}^2(t)$. Taking the derivative of the Lyapunov function gives

$$\begin{aligned}\dot{V}_{2i}(t) &= \dot{V}_{1i}(t) + \sum_{i=1}^{N} z_{2i}(t)\dot{z}_{2i}(t)\\ &= \dot{V}_{1i}(t) + \sum_{i=1}^{N} z_{2i}(t)\big(-g + g_{4i}(t)u_{1i}(t) - \dot{\alpha}_{1i}(t)\big).\end{aligned} \quad (25)$$

Replacing $u_{1i}(t)$ from Eq. 16 into 25 gives

$$\begin{aligned}\dot{V}_{2i}(t) &= -\sum_{i=1}^{N} k_{1i}e_{1i}^2(t) + \sum_{i=1}^{N} A_{1i}(t)z_{2i}(t)\\ &\quad + \sum_{i=1}^{N} z_{2i}(t)\big(-g - \dot{\alpha}_{1i}(t) + g + \dot{\alpha}_{1i}(t) - A_{1i}(t) - k_{2i}z_{2i}(t)\big)\\ &= -\sum_{i=1}^{N} k_{1i}z_{1i}^2(t) - \sum_{i=1}^{N} k_{2i}z_{2i}^2(t).\end{aligned} \quad (26)$$

Therefore, one can conclude from Eq. (26) that $\dot{V}_{2i}(t) < 0$, which completes the proof.

4.2 Translational subsystems

Herein, the virtual controllers for the translational subsystems presented in Eq. (6) for the *x* and *y* coordinates are formulated in accordance with Theorem 2.

Theorem 2: Assume that the translational subsystems of the *i*th quadcopter in a fleet of N quadcopters can be described by Eq. (6) in the *x* and *y* directions using the virtual control inputs defined in Eqs. (7) and (8). Then, the virtual control

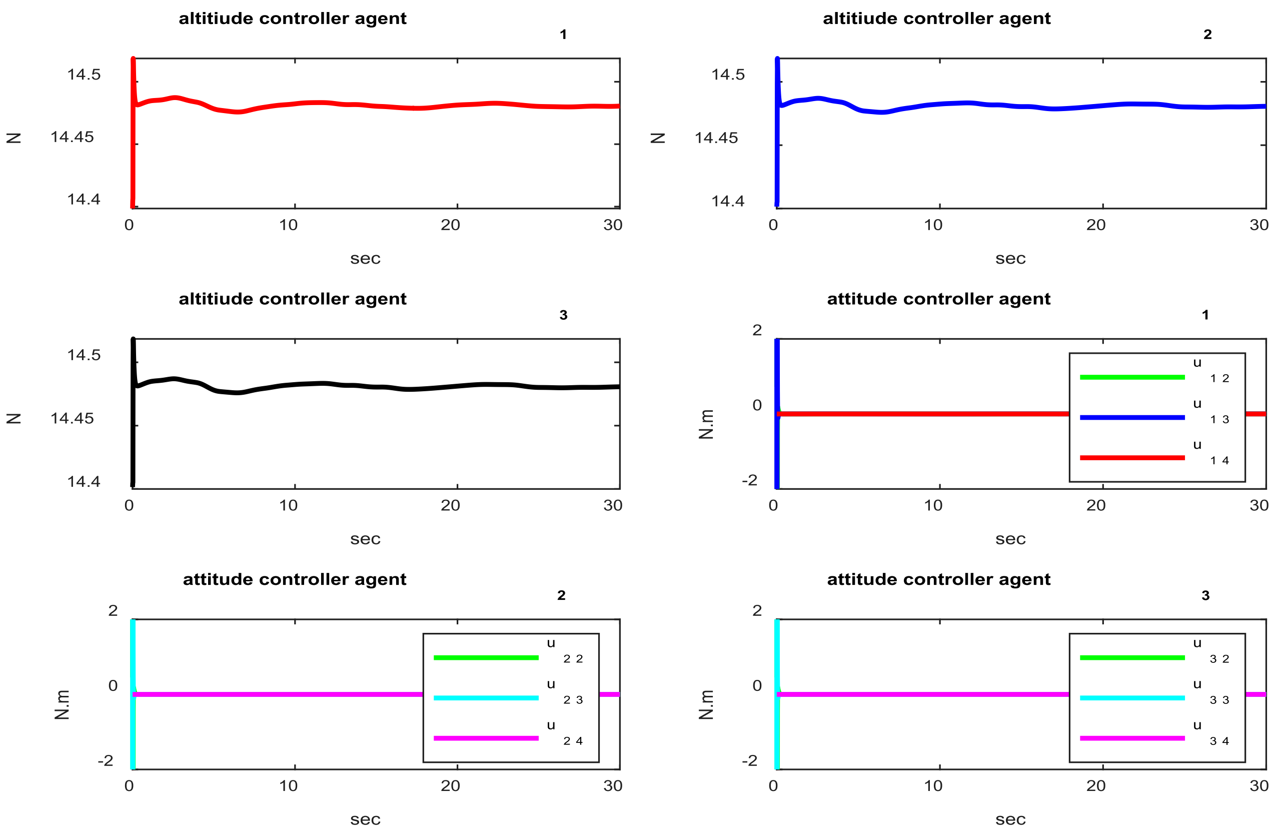

FIGURE 6
Control signals of the attitude subsystem.

inputs $u_{iv3}(t)$ and $u_{iv5}(t)$ for the ith quadcopter can be designed as

$$
\begin{aligned}
u_{iv3}(t) &= g_{4i}^{-1}(t)\left[\dot{\alpha}_{3i}(t) - A_{3i}(t) - k_{4i}z_{4i}(t)\right],\\
u_{iv5}(t) &= g_{6i}^{-1}\left[\dot{\alpha}_{5i}(t) - A_{5i}(t) - k_{6i}z_{6i}(t)\right],
\end{aligned}
\tag{27}
$$

where

$$
\begin{aligned}
\alpha_{3i}(t) &= \dot{x}_{3id}(t) + \frac{-k_{3i}e_{3i}^{2}(t)}{D_{3i}(t)e_{3i}(t) - 2\sum\limits_{j\in N_i}^{N} B_{3ij}(t)e_{3j}(t)},\\
A_{3i}(t) &= D_{3i}(t)e_{3i}(t) - 2\sum_{j\in N_i}^{N} B_{3ij}(t)e_{3j}(t),\\
D_{3i}(t) &= B_{3i}(t) + 2\sum_{j\in N_i}^{N} B_{3ij}(t),\\
\alpha_{5i}(t) &= \dot{x}_{5id}(t) + \frac{-k_{5i}e_{5i}^{2}(t)}{D_{5i}(t)e_{5i}(t) - 2\sum\limits_{j\in N_i}^{N} B_{5ij}(t)e_{5j}(t)},\\
A_{5i}(t) &= D_{5i}(t)e_{5i}(t) - 2\sum_{j\in N_i}^{N} B_{5ij}(t)e_{5j}(t),\\
D_{5i}(t) &= B_{5i}(t) + 2\sum_{j\in N_i}^{N} B_{5ij}(t),
\end{aligned}
\tag{28}
$$

with $B_{3i}(t) = \frac{\Omega_{3idH}^{3}}{(\Omega_{3idH} - d_{3ie}(t))^{3}}$, $B_{3ij}(t) = \frac{\Omega_{3ijH}^{3}}{\left(\Omega_{3ijH} - d'_{3ije}(t)\right)^{3}}$, $B_{5i}(t) = \frac{\Omega_{5idH}^{3}}{(\Omega_{5idH} - d_{5ie}(t))^{3}}$, $B_{5ij}(t) = \frac{\Omega_{5ijH}^{3}}{\left(\Omega_{5ijH} - d'_{5ije}(t)\right)^{3}}$, $z_{4i}(t) = x_{4i}(t) - \alpha_{3i}(t)$, $z_{6i}(t) = x_{6i}(t) - \alpha_{5i}(t)$, $k_{3i} > 0, k_{4i} > 0$, $k_{5i} > 0$, $k_{6i} > 0$, $\Omega_{1ijH} > 0$, and $\Omega_{1idH} > 0$.

Then, the translational tracking error and interagent collision avoidance conditions are guaranteed by remaining within the sets defined by Eqs. (11), (14), and (15) if the quadcopter starts with the initial conditions such that the tracking errors remain within the same sets, i.e., $d_{3ie}(0) < \Omega_{3idH}$, $d_{5ie}(0) < \Omega_{5idH}$, $d'_{3ije}(0) < \Omega_{3ijH}$, and $d'_{5ije}(0) < \Omega_{5ijH}$, respectively.

Proof: A proof similar to that of Theorem 1 can be considered here and has been omitted for brevity.

5 Proposed attitude control system

Problem 3 is considered in this section, and a BLF-based controller is designed for the attitude subsystem with the dynamics presented in Eq. (9). First, according to Eqs. (7) and (8) as well as the virtual controllers designed for the translational subsystems in Eqs (27) and (28) the desired angles for the roll ($\phi_{id}(t)$) and pitch ($\theta_{di}(t)$) are computed in Eq. (29) as follows:

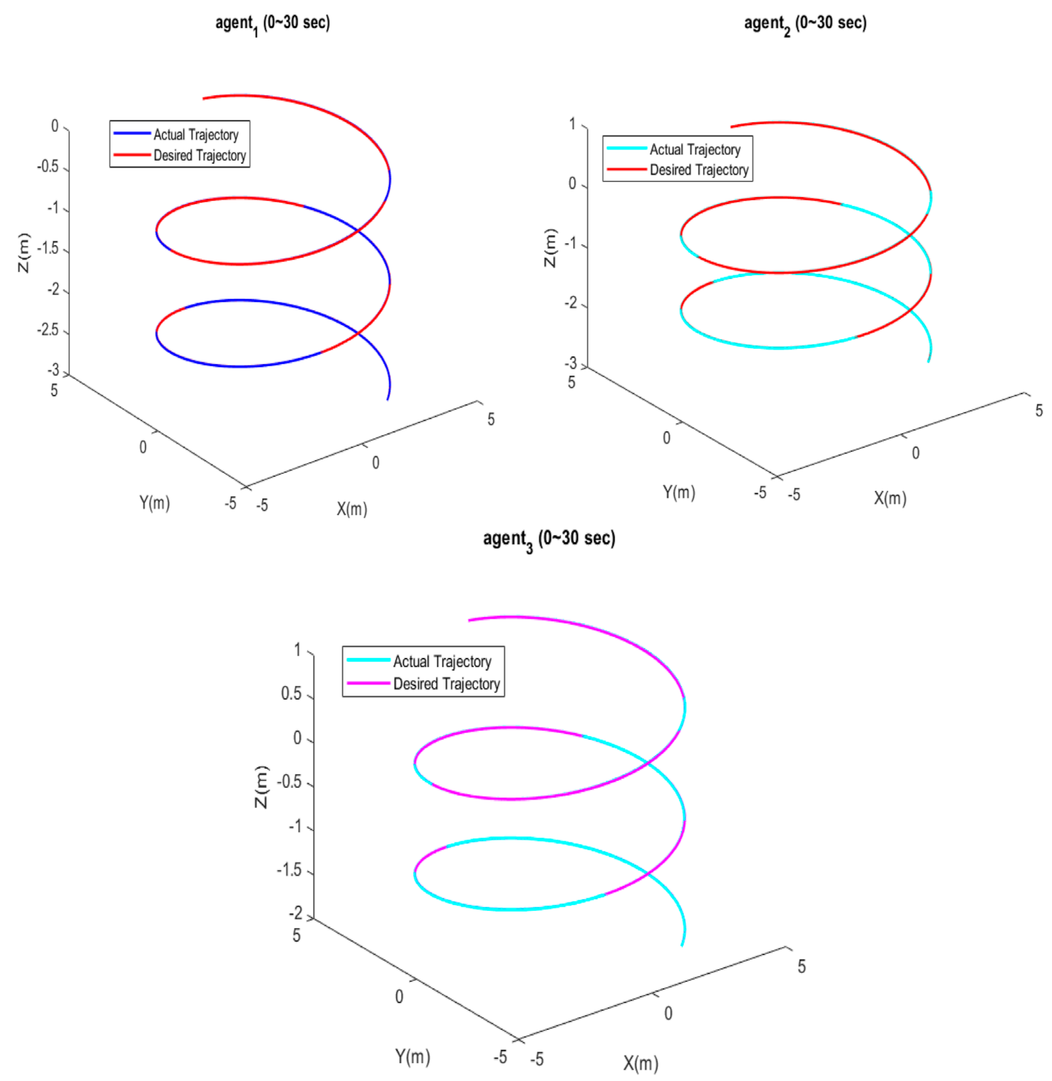

FIGURE 7
Actual and desired 3D positions of the quadcopters.

$$\begin{aligned}\sin\left(\phi_{id}(t)\right) &= u_{iv3}(t)\sin\left(\psi_i(t)\right) - u_{iv5}(t)\cos\left(\psi_i(t)\right)\\ \sin\left(\theta_{di}(t)\right) &= \frac{u_{iv3}(t)\cos\left(\psi_i(t)\right) + u_{iv5}(t)\sin\left(\psi_i(t)\right)}{\cos\left(\phi_{id}(t)\right)}\end{aligned} \tag{29}$$

The desired yaw angle $(\psi_{id}(t))$ can be set freely.

Theorem 3: Assume that the attitude subsystem of the ith quadcopter in a fleet of N quadcopters can be described by Eq. (9); then, the control input vector $u_i(t)$ for the ith quadcopter can be designed as

$$u_i(t) = G_{8i}^{-1}\left(-f_{2i}(t) + \dot{\alpha}_{7i}(t) - B_{7i}(t)e_{7i}(t) - k_{8i}z_{8i}(t)\right), \tag{30}$$

where $z_{8i}(t) = x_{8i}(t) - \alpha_{8i}(t)$ and $\alpha_{8i}(t) = \dot{x}_{7id} - \left(k_{7i}e_{7i}(t)/B_{7i}(t)\right)$, with $B_{7i} = \frac{\Omega_{7idH}^3}{(\Omega_{7idH} - d_{7ie})^3}$ and k_{7i}, k_{8i}, and Ω_{7idH} being scalar positive constants.

Then, the attitude tracking error is guaranteed by remaining within the set defined in Eq. (12) if the quad starts from the initial conditions such that the tracking errors remain within the same sets, i.e., $d_{7ie}(0) = \|e_{7i}(0)\| < \Omega_{7idH}$.

Proof: We consider the BLF $V_{7i}(t) = \frac{1}{2}\eta_{7ie}^2(t)$ with $\eta_{7ie}(t) = \frac{\Omega_{7idH}d_{7ie}(t)}{\Omega_{7idH} - d_{7ie}(t)}$; therefore,

$$\begin{aligned}\dot{V}_{7i}(t) &= \frac{\Omega_{7idH}^3}{\left(\Omega_{7idH} - d_{7i}(t)\right)^3} e_{7i}^T(t)\dot{e}_{7i}(t)\\ &= B_{7i}(t)e_{7i}^T(t)\dot{e}_{7i}(t).\end{aligned} \tag{31}$$

Since $\dot{e}_{7i}(t) = x_8(t) - \dot{x}_{7id}(t) = z_{8i}(t) + \alpha_{8i}(t) - \dot{x}_{7id}(t)$, if we select $\alpha_{8i}(t) = \dot{x}_{7id} - \left(k_{7i}e_{7i}(t)/B_{7i}(t)\right)$,we obtain $\dot{V}_{7i}(t) = B_{7i}(t)e_{7i}^T(t)z_{8i}(t) - k_{7i}e_{7i}^T(t)e_{7i}(t)$. Now, a Lyapunov function is chosen by adding a quadratic function to V_{7i} as follows:

$$V_{8i}(t) = V_{7i}(t) + \frac{1}{2}z_{8i}^T(t)z_{8i}(t). \tag{32}$$

Therefore, using Eq. (32) one can conclude Eq. (33) as follows:

$$\begin{aligned}\dot{V}_{8i}(t) &= \dot{V}_{7i}(t) + z_{8i}^T(t)\dot{z}_{8i}(t)\\ &= B_{7i}(t)e_{7i}^T(t)z_{8i}(t) - k_{7i}e_{7i}^T(t)e_{7i}(t)\\ &\quad + z_{8i}^T(t)\left(\dot{x}_{8i}(t) - \dot{\alpha}_{8i}(t)\right).\end{aligned} \tag{33}$$

Now, according to Eq. (9), by selecting $u_i(t)$ as Eq. (30) and using Eq. (31) it is concluded that $\dot{V}_{8i}(t) = -k_{7i}e_{7i}^T(t)e_{7i}(t) - k_{8i}z_{8i}^T(t)z_{8i}(t) < 0$. Therefore, the attitude tracking objective in Eq. (12) is satisfied if $d_{7ie}(0) = \|e_{7i}(0)\| < \Omega_{7idH}$, hence completing the proof.

Figure 1 depicts the general structure of the proposed controller for the ith agent. The overall quadcopter system has three subsystems. The design of $u_{1i}(t)$ starts from the altitude subsystem. Then, this controller is used to design the virtual controllers in the translational subsystems. Finally, the control inputs of the attitude subsystem are designed to meet the desired control objectives. Owing to the fact that the graph topology of the quadcopter system is connected, the neighboring information is used to achieve safety, collision avoidance, and stability.

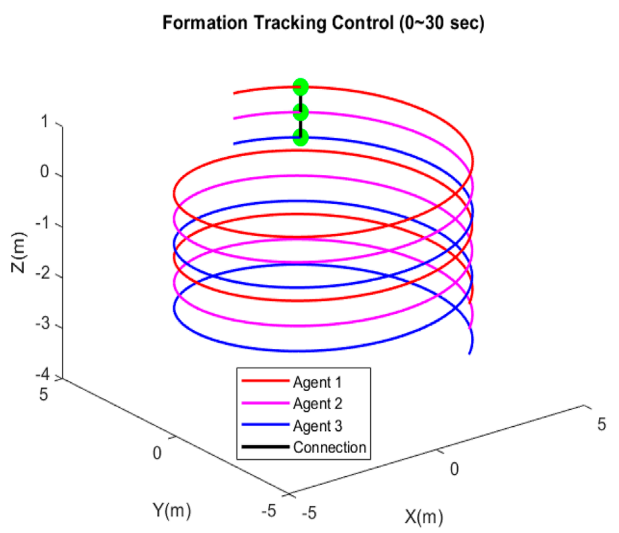

FIGURE 8
Three-dimensional formation tracking control and formation shape.

6 Simulation results

In this section, simulation results are provided to demonstrate the efficiency of the proposed method. Figure 2 depicts the interconnection of three quadcopters considered for the simulation.

The initial conditions are considered as follows: $x_{11}(0) = 0, x_{12}(0) = 0.5, x_{13}(0) = 1$, $x_{21}(0) = x_{22}(0) = x_{23}(0) = -0.08$, $x_{31}(0) = x_{32}(0) = x_{33}(0) = 0$, $x_{41}(0) = x_{42}(0) = x_{43}(0) = 1.5$, $x_{51}(0) = x_{52}(0) = x_{53}(0) = 3.9$, $x_{61}(0) = x_{62}(0) = x_{63}(0) = 0$, $x_{7i}(0) = [0 \quad 0 \quad 0], i = 1,2,3$, and $x_{8i}(0) = [0 \quad 0 \quad 0], i = 1,2,3$. The physical parameters of the quadcopters are as follows: $m_i = 1.47kg$, $I_{xxi} = I_{yyi} = 0.01152\,kgm^2$, $I_{zzi} = 0.0218\,kgm^2$, and $L_i = 0.28$, $i = 1,2,3$. The reference trajectory for the movement of the quadcopters is given in Eq. (34) as follows:

$$\begin{aligned}&z_{1d}(t) = -0.1t, z_{2d}(t) = -0.1t + 0.5, z_{3d}(t) = -0.1t + 1\\ &x_{1d}(t) = x_{2d}(t) = x_{3d}(t) = 4\sin(0.5t)\\ &y_{1d}(t) = y_{2d}(t) = y_{3d}(t) = 4\cos(0.5t).\end{aligned} \tag{34}$$

As mentioned previously, the values $\Omega_{1idH} > 0$ to $\Omega_{1=7idH} > 0$ are the upper limits for distance tracking errors d_{1ie} to d_{7ie}, while $\Omega_{1ijH} > 0$, $\Omega_{3ijH} > 0$, and $\Omega_{5ijH} > 0$ are the respective upper limits for the distance tracking errors d'_{1ije}, d'_{3ije}, and d'_{5ije}. These two sets of parameters determine the safe sets for the movements of the quadcopters. If these values are selected to be large, although the safety set will be larger, it may cause problems for the system in terms of safety as a wider range of errors would be considered acceptable. If these values are too small, then the safe set will be too small and forces the selection of the initial values to be very close to the real ones, which is unrealistic and may force the algorithm to be very sensitive to small deviations of the errors. Therefore, the selection of these two sets of parameters is very important. In the simulations, they are selected as follows: $\Omega_{7idH} = 1.5$, $\Omega_{1idH} = \Omega_{1ijH} = 0.1$, $\Omega_{3idH} = \Omega_{3ijH} = 0.49$, and $\Omega_{5idH} = \Omega_{5ijH} = 0.25$. The simulation results are as follows.

Figure 3 shows the distances between the agents, indicating that the agents are collision-free and maintain distances specified by the reference trajectories between the quadcopters during movement. According to Figure 4, it is clear that the attitude control subsystem is well designed as the states (ϕ, θ, ψ) follow the desired trajectories. The position tracking errors of the agents in the x, y, and z axes are depicted in Figure 5, according to which the error is less than 0.03; this shows that the controller is well designed and that the tracking error is acceptable. It is clear from Figure 6 that the value of the control signal $u_{1i}, i = 1,2,3$ converges approximately to 14.48 N and that the values of the control signals for the attitude subsystem converge to 0 N·m.

Figure 7 depicts that each quadcopter follows its desired path; thus, each quadcopter tracks its desired trajectory successfully during flight. The results of formation tracking as well as the formation shape are depicted in Figure 8; it is obvious from the figure that the quadcopters follow their trajectories in formation without any collisions and that the desired distances between them are maintained. Figure 9 shows that the error value converges to a constant equal to 0.0083, with a settling time of 50 s.

The method proposed in this work is compared with that of Kuriki and Namerikawa (2015) in Table 1, from which it is obvious that the proposed method is significantly superior based on different aspects. The root mean-squared error (RMSE) as well as steady-state error values for the proposed method are considerably lower, and our method is significantly faster. Although the settling time in our method is a bit large, it still outperforms the oscillating behavior of the method proposed by Kuriki and Namerikawa (2015).

7 Conclusion

The purpose of this work was to design a distributed collision-free formation tracking control scheme for multiquadcopter systems using the BLF in a backstepping procedure. The controllers were designed in a hierarchical structure to tackle the underactuated nature of the quadcopter system. Accordingly, the altitude controller was designed first, followed by the translational controller with virtual inputs. The desired Euler angles were then obtained using

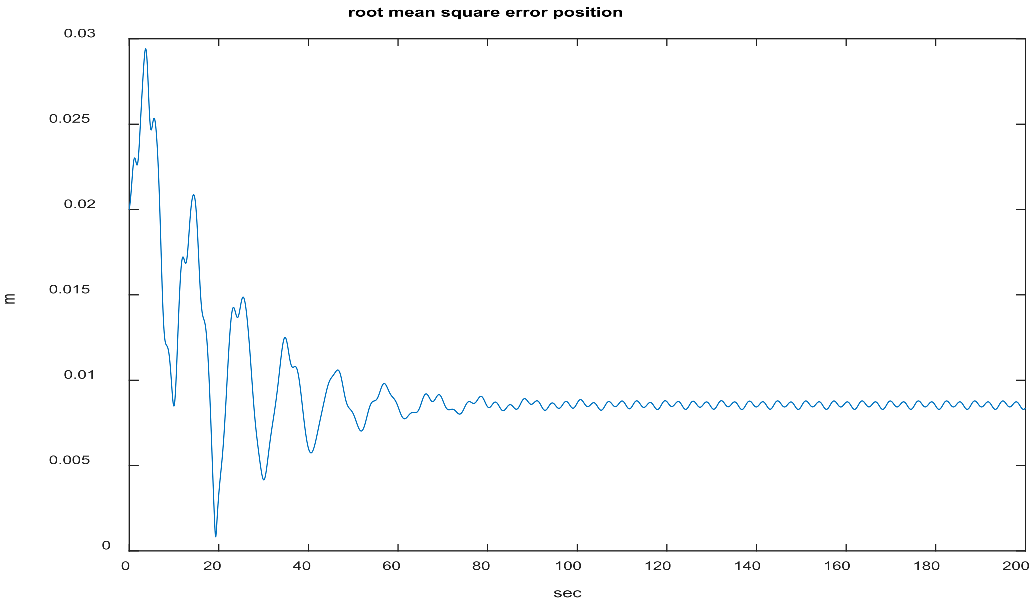

FIGURE 9
Root mean-squared error (RMSE) between the desired and actual positions of the agents.

TABLE 1 Comparison of the proposed method with the system of Kuriki and Namerikawa (2015).

	Settling time (s)	RMSE (m)	Average calculation time (s)	Steady-state error (m)
Proposed method	50	0.0083	0.000053741	0.0083
Kuriki and Namerikawa (2015)	Fluctuating behavior	0.2374	0.005	0.2288

the virtual control signals and were finally employed to derive the proposed BLF-based attitude control subsystem. Simulations were performed to demonstrate the control objectives designed and achieved herein, including safety (staying in a safe set) and collision avoidance as well as formation tracking control. By adding the uncertainty terms and noise to the dynamics of the system, the controller can be designed such that it meets the control goals when the specified cases occur; this can be considered as a suggestion for future work. Formulating the problem of obstacle avoidance using the BLF is also suggested as a future work.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary material, and any further inquiries may be directed to the corresponding authors.

Author contributions

MS: Investigation, Software, Visualization, Writing–original draft. NS-N: Methodology, Project administration, Supervision, Writing–review and editing. RB: Project administration, Writing–review and editing, Investigation, Validation. AM: Validation, Writing–review and editing, Funding acquisition, Methodology, Resources, Supervision.

Funding

The author(s) declare that no financial support was received for the research, authorship, and/or publication of this article.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The author(s) declare that they were an editorial board member of Frontiers at the time of submission. This had no impact on the peer review process or the final decision.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations or those of the publisher, editors, and reviewers. Any product that may be evaluated in this article or claim that may be made by its manufacturer is not guaranteed or endorsed by the publisher.

References

Ahmadi, K., Asadi, D., Nabavi-Chashmi, S.-Y., and Tutsoy, O. (2023). Modified adaptive discrete-time incremental nonlinear dynamic inversion control for quad-rotors in the presence of motor faults. *Mech. Syst. Signal Process.* 188, 109989. doi:10.1016/j.ymssp.2022.109989

Allahyar, S. D. W. Z. M. (2023). "Robotics and artificial intelligence in the nuclear industry: from teleoperation to cyber physical systems," in *Artificial intelligence for robotics and autonomous systems applications A*. Editor T. A. A. Koubaa (Berlin, Germany: Springer), 123–166.

Burrell, T., West, C., Monk, S. D., Montezeri, A., and Taylor, C. J. (2018). "Towards a cooperative robotic system for autonomous pipe cutting in nuclear decommissioning," in 2018 UKACC 12th International Conference on Control (CONTROL), Sheffield, UK, September, 2018. doi:10.1109/control.2018.8516841

Can, A., Imran, I. H., Price, J., and Montazeri, A. (2022). Robust formation control and trajectory tracking of multiple quadrotors using a discrete-time sliding mode control technique. *IFAC-PapersOnLine* 55 (10), 2974–2979. doi:10.1016/j.ifacol.2022.10.184

Chen, Y., Singletary, A., and Ames, A. D. (2020). Guaranteed obstacle avoidance for multi-robot operations with limited actuation: a control barrier function approach. *IEEE Control Syst. Lett.* 5 (1), 127–132. doi:10.1109/lcsys.2020.3000748

Ganguly, S. (2022). Robust trajectory tracking and payload delivery of a quadrotor under multiple state constraints. Available at: https://arxiv.org/abs/2201.03711.

Hu, J., Bhowmick, P., Jang, I., Arvin, F., and Lanzon, A. (2021). A decentralized cluster formation containment framework for multirobot systems. *IEEE Trans. Robotics* 37 (6), 1936–1955. doi:10.1109/tro.2021.3071615

Imran, I. H., and Montazeri, A. (2022). Distributed robust synchronization control of multiple heterogeneous quadcopters with an active virtual leader. *IFAC-PapersOnLine* 55 (10), 2659–2664. doi:10.1016/j.ifacol.2022.10.111

James, M., Carr, B., D'Arcy, F., Diefenbach, A., Dietterich, H., Fornaciai, A., et al. (2020). Volcanological applications of unoccupied aircraft systems (UAS): developments, strategies, and future challenges. *Volcanica* 3, 67–114. doi:10.30909/vol.03.01.67114

Jin, X., Dai, S.-L., Liang, J., and Guo, D. (2021). Multirobot system formation control with multiple performance and feasibility constraints. *IEEE Trans. Control Syst. Technol.* 30 (4), 1766–1773. doi:10.1109/tcst.2021.3117487

Julian, K. D. K. M. J., and Kochenderfer, M. J. (2019). Distributed wildfire surveillance with autonomous aircraft using Deep reinforcement learning. *J. Guid. Control, Dyn.* 48 (2), 1768–1778. doi:10.2514/1.g004106

Khadhraoui, A., Zouaoui, A., and Saad, M. (2023). Barrier Lyapunov function and adaptive backstepping-based control of a quadrotor UAV. *Robotica* 41 (10), 2941–2963. doi:10.1017/s0263574723000735

Kumar, S., and Kumar, S. R. (2022). "Barrier lyapunov-based nonlinear trajectory following for unmanned aerial vehicles with constrained motion," in 2022 International Conference on Unmanned Aircraft Systems (ICUAS), Dubrovnik, Croatia, June, 2022. doi:10.1109/icuas54217.2022.9836037

Kuriki, Y., and Namerikawa, T. (2015). Formation control with collision avoidance for a multi-UAV system using decentralized MPC and consensus-based control. *SICE J. Control, Meas. Syst. Integration* 8 (4), 285–294. doi:10.9746/jcmsi.8.285

Liang, Y., Qi, D., and Yanjie, Z. (2020). Adaptive leader–follower formation control for swarms of unmanned aerial vehicles with motion constraints and unknown disturbances. *Chin. J. Aeronautics* 33 (11), 2972–2988. doi:10.1016/j.cja.2020.03.020

Liu, Y., and Bucknall, R. (2018). A survey of formation control and motion planning of multiple unmanned vehicles. *Robotica* 36 (7), 1019–1047. doi:10.1017/s0263574718000218

Lizzio, F. F., Capello, E., and Guglieri, G. (2022). A review of consensus-based multi-agent UAV implementations. *J. Intelligent Robotic Syst.* 106 (2), 43. doi:10.1007/s10846-022-01743-9

Mansfield, D., and Montazeri, A. (2024). A survey on autonomous environmental monitoring approaches: towards unifying active sensing and reinforcement learning. *Front. Robotics AI* 11, 1336612. doi:10.3389/frobt.2024.1336612

Martin, P. G., Kwong, S., Smith, N., Yamashiki, Y., Payton, O., Russell-Pavier, F., et al. (2016). 3D unmanned aerial vehicle radiation mapping for assessing contaminant distribution and mobility. *Int. J. Appl. Earth Observation Geoinformation* 52, 12–19. doi:10.1016/j.jag.2016.05.007

Montazeri, A., Can, A., and Imran, I. H. (2021). "Chapter 3 - unmanned aerial systems: autonomy, cognition, and control," in *Unmanned aerial systems.* Editors A. Koubaa, and A. T. Azar (Cambridge, Massachusetts, United States: Academic Press), 47–80.

Mughees, A., and Ahmad, I. (2023). Multi-optimization of novel conditioned adaptive barrier function integral terminal SMC for trajectory tracking of a quadcopter System. *IEEE Access* 11, 88359–88377. doi:10.1109/access.2023.3304760

Neumann, T. F. A., Kallweit, S., and Scholl, I., Towards a mobile mapping robot for underground mines. November 2014; pp. In Proceedings of the 2014 PRASA, RobMech and AfLaT International Joint Symposium Cape Town, South Africa, 2014: p. 27–28.

Ngo, K. B., Mahony, R., and Jiang, Z.-P. (2005). "Integrator backstepping using barrier functions for systems with multiple state constraints," in Proceedings of the 44th IEEE Conference on Decision and Control, Seville, Spain, December, 2005.

Patil, A., and Shah, G. (2021). "Discrete time consensus algorithm in multi-agent system," in 2021 Seventh Indian Control Conference (ICC), Mumbai, India, December, 2021. doi:10.1109/icc54714.2021.9702911

Peng, X., Wang, Q., and Xiong, S. (2020). Distributed leader-follower consensus tracking control for fixed-wind uavs with positive linear speeds under directed graphs. *IFAC-PapersOnLine* 53 (5), 487–490. doi:10.1016/j.ifacol.2021.04.134

Sadeghzadeh-Nokhodberiz, N., Can, A., Stolkin, R., and Montazeri, A. (2021). Dynamics-based modified fast simultaneous localization and mapping for unmanned aerial vehicles with joint inertial sensor bias and drift estimation. *IEEE Access* 9, 120247–120260. doi:10.1109/access.2021.3106864

Sadeghzadeh-Nokhodberiz, N., Iranshahi, M., and Montazeri, A. (2023). Vision-based particle filtering for quad-copter attitude estimation using multirate delayed measurements. *Front. Robotics AI* 10, 1090174. doi:10.3389/frobt.2023.1090174

Sadeghzadeh-Nokhodberiz, N., and Meskin, N. (2023). Consensus-based distributed Formation Control of multi-quadcopter systems: barrier lyapunov function approach. *IEEE Access* 11, 142916–142930. doi:10.1109/access.2023.3340417

Tang, Z.-L., Tee, K. P., and He, W. (2013). Tangent barrier Lyapunov functions for the control of output-constrained nonlinear systems. *IFAC Proc. Vol.* 46 (20), 449–455. doi:10.3182/20130902-3-cn-3020.00122

Tee, K. P., and Ge, S. S. (2011). Control of nonlinear systems with partial state constraints using a barrier Lyapunov function. *Int. J. Control* 84 (12), 2008–2023. doi:10.1080/00207179.2011.631192

Tee, K. P., Ge, S. S., and Tay, E. H. (2009). Barrier Lyapunov functions for the control of output-constrained nonlinear systems. *Automatica* 45 (4), 918–927. doi:10.1016/j.automatica.2008.11.017

Tee, K. P., Ge, S. S., and Tay, F. E. H. (2008). Adaptive control of electrostatic microactuators with bidirectional drive. *IEEE Trans. control Syst. Technol.* 17 (2), 340–352. doi:10.1109/TCST.2008.2000981

Yan, J., Guan, X., Luo, X., and Chen, C. (2017). Formation control and obstacle avoidance for multi-agent systems based on virtual leader-follower strategy. *Int. J. Inf. Technol. Decis. Mak.* 16 (03), 865–880. doi:10.1142/s0219622014500151

Yasin, J. N., Mohamed, S. A. S., Haghbayan, M. H., Heikkonen, J., Tenhunen, H., and Plosila, J. (2020). Unmanned aerial vehicles (uavs): collision avoidance systems and approaches. *IEEE access* 8, 105139–105155. doi:10.1109/access.2020.3000064