IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 19, 2020, accepted March 30, 2020, date of publication April 20, 2020, date of current version June 3, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2988773

Comparison of Linear and Nonlinear Methods for Distributed Control of a Hierarchical Formation of UAVs

ANAM TAHIR[1], JARI M. BÖLING[2], MOHAMMAD-HASHEM HAGHBAYAN[1], (Member, IEEE), AND JUHA PLOSILA[1], (Member, IEEE)

[1] Autonomous Systems Laboratory, Department of Future Technologies, Faculty of Science and Engineering, University of Turku, 20014 Turku, Finland
[2] Laboratory of Process and Systems Engineering, Åbo Akademi University, 20500 Turku, Finland

Corresponding author: Anam Tahir (anam.tahir@utu.fi)

This work was supported in part by the Academy of Finland, project no. 314048.

ABSTRACT A key problem in cooperative robotics is the maintenance of a geometric configuration during movement. As a solution for this, a multi-layered and distributed control system is proposed for the swarm of drones in the formation of hierarchical levels based on the leader–follower approach. The complexity of developing a large system can be reduced in this way. To ensure the tracking performance and response time of the ensemble system, nonlinear and linear control designs are presented; (a) Sliding Mode Control connected with Proportional-Derivative controller and (b) Linear Quadratic Regular with integral action respectively. The safe travel distance strategy for collision avoidance is introduced and integrated into the control designs for maintaining the hierarchical states in the formation. Both designs provide a rapid adoption with respect to their settling time without introducing oscillations for the dynamic flight movement of vehicles in the cases of (a) nominal, (b) plant-model mismatch, and (c) external disturbance inputs. Also, the nominal settling time of the swarm is improved by 44% on average when using the nonlinear method as compared to the linear method. Furthermore, the proposed methods are fully distributed so that each UAV autonomously performs the feedback laws in order to achieve better modularity and scalability.

INDEX TERMS Unmanned aerial vehicles (UAVs), distributed control, hierarchical systems.

I. INTRODUCTION

Recently, the use of swarms/fleets of Unmanned Aerial Vehicles (UAVs), or drones, have become increasingly popular; attracting attention from researchers and industries representing miscellaneous disciplines [1]. Controlling the formation of a swarm using either centralized or decentralized (i.e. distributed) architecture is a demanding control problem. Due to the lack of observability, each drone might have to dynamically adjust its location w.r.t. the other drones.

Based on recent studies, the strategies to control the formation of a drone swarm are generally divided into three categories; (a) leader–follower [2]–[4], (b) virtual structure [5]–[7], and (c) behavioural [8]–[10] approaches. In a leader–follower approach, one or more drones are given a leading role, i.e. the responsibility to decide on the general path of the swarm. The desired positions of the other drones, the followers, are defined relative to the actual states of the associated leaders. A swarm can be further divided into subgroups of leader–followers, building a hierarchy, which improves the scalability of the approach. The main advantage of this approach is that the path planning algorithm needs to be executed only in the leaders' workspace which significantly reduces the overall computation time of path planning, indicating fast decision making within the swarm. In a virtual structure approach, a virtual moving structure, i.e. a formation, decides the pose reference of each element of the fleet. Based on these pose references, the controller manipulates the actuators to reduce the error as much as possible. In this approach, each virtual vacant pose can be filled by any drone, and therefore, an actual leader is not needed. In contrast to the first two, in a behaviour-based approach, the actuation is defined as several desired behaviours that are assigned to each drone. The overall control is derived by assigning different weights to behaviours in each drone to form the desired shape of the swarm.

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

One of the main problems in distributed control of the formation is the lack of perception of the dynamics of the whole swarm in each partially distributed controller that is being executed in each drone of the swarm, i.e. *top–down approach*. In most of the state-of-the-art focusing on distributed controllers, the local controller of each drone, unaware of the behaviour and constraints of the swarm's higher levels of hierarchy, only manipulates its actuators based on its observations and local dynamic models of a subset of all nodes constituting the swarm, i.e. *bottom–up approach*. These techniques try to optimize and refine the partial behaviour of the system that subsequently contributes to general emergent behaviour. However, manipulating only the actuators in each node based on partial dynamic models cannot guarantee optimal or even near-optimal desired top-level behaviour of the system, which is the final goal of such control design. Therefore, this work aims to propose a strategy to reach this goal. In other words, it addresses the problem based on the overall dynamics of the whole swarm and the local dynamics of a node in order to implement the partial controller of each drone.

The main contributions of this paper can be summarized as follows:

- Reducing the complexity of the large system of swarming drones by implementing the decentralized control systems using leader–follower architecture in which the goal is to form the hierarchical states for the swarm of drones that keep the desired formation using setpoints/offsets to avoid collisions with each other.
- Testing of distributed control designs based on Sliding Mode Control (SMC) with Proportional-Derivative (PD) and Linear Quadratic Regulator (LQR) with integral action to guarantee the tracking performance in such a way that the system's settling time is reduced as much as possible without introducing oscillations in the response.
- Presenting the side by side comparative analysis based on setpoint tracking, response and settling time, collision avoidance, and computational costs for the proposed distributed control systems. Also, the testing of plant-model mismatch and external disturbance inputs are examined.

To do this, two well-known control methods, i.e. SMC PD and LQR integral action, are adapted and integrated that are applied to each drone of a hierarchical swarm formation. Both control methods are responsible to fulfil the high-level expectation from the hierarchical swarm formation, e.g. translational positioning, and to fulfil the low-level expectation, e.g. the stability of angular positioning. More precisely, in LQR integral action, both are done using a single constant state feedback law. While in SMC PD, the overall control system of each drone in a swarm is composed of two main parts, i.e. inner and outer feedback loops. The inner loop is responsible for angular positions of the drone while the outer loop is responsible for its translational positions. The outer loop observes the current translational pose of the drone and tries to eliminate the lateral, longitudinal, and vertical distance errors and generates a reference for pitch, roll, and yaw angles respectively to be tracked by the inner loop that specifies engine propulsion. The parameters of the controllers change based on the location of each drone in the swarm. This provides a flexible reconfigurable grid of sub-functionalities demonstrated by individual drones contributing to the general behaviour of the swarm. Each of the sub-functionalities can be adjusted according to the expected behaviour from the located drone and via manipulating the controller's parameters.

This paper is divided into seven sections. The motivation and problem formulation is stated in section I. Related work is discussed in section II. The dynamics of the model for the considered system is presented in section III. The control designing for the swarm of drones in a hierarchical manner is described in section IV. The simulation setup and the performance analysis are elaborated in section V and VI respectively. Lastly, concluding remarks are given in section VII.

II. RELATED WORK

In [11], Proportional Integral Derivative (PID) control as the inner loop and parametric weighted control as the outer loop of the controller for a swarm of four identical drones are investigated. By changing a few parameters in the parametric weighed controller using four control input drivers, it is shown that the behaviour of UAVs can be rapidly changed in-field testing. However, each node has its own histogram of visited areas, and synchronized information does not get shared to help the swarm as a whole. In [12], PID control for the inner loop and extended nonlinear dynamic inversion control for the outer loop in the context of autonomous formation flight of three drones are proposed. Regardless of ambiguity in response time efficiency and error measurements of the system, the scalability and top–down/bottom–up influence of dynamics on the controller are not comprehensively considered. In [13], for hovering synchronization in horizontal and vertical planes using three drones, PD control as the inner loop and multi-agent-based consensus control as the outer loop are presented. A Type 2 system is considered in order to perform the simulation tests. In [14], experiments have been conducted to study the flight test for a quadcopter using a PD control architecture. Also, the swarm behaviour through a coordinated flight path for a leader drone with four follower drones is highlighted. In [15], an event-triggered distributed Model Predictive Control (MPC) technique for formation control of three drones is studied, where each drone shares information with its neighbours. In [16], a fuzzy logic approach for task assignment and controlling a swarm of agents, representing UAVs, in a decentralized environment is examined, and the results of collision attempts and completion times between the distributed and centralized strategies through simulations are compared. The central control design is based on finite state machines, and the proposed solution is simulated using the Unity3D engine. Furthermore, in [17], SMC and integral SMC for nonlinear second-order multi-agent systems with unknown nonlinear interactions are presented. However, the method lacks scalability in case of increasing number of drones. In [18] and [19], a performance analysis is considered, in a centralized environment with three drones, using SMC for the inner control loop combined with LQR control for the outer control loop. However, in [18], the results for the proposed architecture lack the fast response time, and in [19], the controllability matrix related to the linear dynamics of the position controller does not have the full row rank. Hence, the given set of equations is not feasible for designing position control of x and y using the proposed method due to the lack of information provided.

III. DYNAMIC MODEL OF A DRONE

In this section, the dynamic model of a leader–follower swarm of drones is examined. This model serves as a case study to demonstrate the idea of the proposed control designs. In the system under consideration, the swarm consists of a hierarchical organization of leader–follower drones that are responsible for tracking the desired trajectories as well as for hovering at desired positions for given time periods. The trajectory of each follower is defined based on the orientation and actions of its respective leader.

The model of each node in the swarm is based on the model of a quadcopter, i.e. a drone that has four propellers with fixed pitch mechanically movable blades, as shown in Fig. 1.

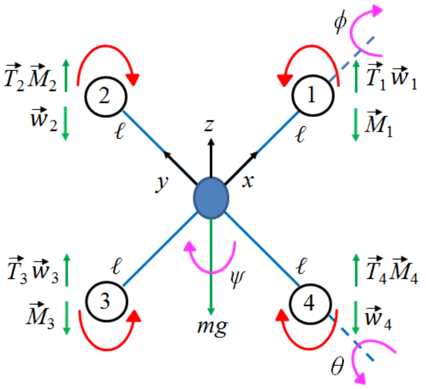


FIGURE 1. Kinematics of the quadcopter.

The major forces acting on the quadcopter are the gravity g and the thrust T_i, $i \in \{1, 2, 3, 4\}$, of the propellers. In this model, the inertial reference is the earth shown as (x, y, z) that is the origin of the reference frame. The drone is assumed to be a rigid body that has the constant mass symmetrically distributed with respect to the planes (x, y), (y, z), and (x, z). The two possible configurations, i.e. cross and plus, are used for most of the quadcopter designs. The cross configuration is considered as more suitable than the plus configuration in case of aerial photography because it better keeps the propellers out of the camera view.

The orientation of a quadcopter reference frame (x, y, z) with respect to an inertial frame $(x, y, z)_0$ can be expressed mathematically in a state variable form [20]–[22], where translational and angular accelerations are given by

$$\begin{aligned}\dot{v}_x &= -v_z w_y + v_y w_z - g\sin\theta\\ \dot{v}_y &= -v_x w_z + v_z w_x + g\cos\theta\sin\phi\\ \dot{v}_z &= -v_y w_x + v_x w_y + g\cos\theta\cos\phi - \frac{T}{m}\end{aligned} \tag{1}$$

and

$$\begin{aligned}\dot{w}_x &= \frac{1}{J_x}(-w_y w_z(J_z - J_y) + M_x - \frac{k_{wT}}{k_{MT}} J_{mp} M_z w_y)\\ \dot{w}_y &= \frac{1}{J_y}(-w_x w_z(J_x - J_z) + M_y - \frac{k_{wT}}{k_{MT}} J_{mp} M_z w_x)\\ \dot{w}_z &= \frac{M_z}{J_z}\end{aligned} \tag{2}$$

respectively. The thrust produced by each propeller T_i is translated into a total thrust T and the reactive torques M_i, $i \in \{x, y, z\}$, is affecting the rotations along the corresponding axis. The J_i, $i \in \{x, y, z\}$, is known as the moment of inertia along the corresponding axis, and J_{mp} is the moment of inertia of a motor with propeller. The angular velocities of propellers are assumed to be proportional to thrusts of propellers, i.e. $w_j = k_{wT} T_i, j \in \{x, y, z\}$. Similarly, the reactive moments of propellers are assumed to be proportional to the thrust of propellers, i.e. $M_i = k_{MT} T_i$. Depending on the chosen configuration, the propeller thrusts T_i will generate different thrust T and torques M_i namely

$$\begin{bmatrix} T \\ M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -\ell & 0 & \ell \\ \ell & 0 & -\ell & 0 \\ k_{MT} & -k_{MT} & k_{MT} & -k_{MT} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} \tag{3}$$

for plus configuration and

$$\begin{bmatrix} T \\ M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{\sqrt{2}}{2}\ell & -\frac{\sqrt{2}}{2}\ell & -\frac{\sqrt{2}}{2}\ell & \frac{\sqrt{2}}{2}\ell \\ \frac{\sqrt{2}}{2}\ell & \frac{\sqrt{2}}{2}\ell & -\frac{\sqrt{2}}{2}\ell & -\frac{\sqrt{2}}{2}\ell \\ k_{MT} & -k_{MT} & k_{MT} & -k_{MT} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} \tag{4}$$

for cross configuration where ℓ is a length of the fixed pitch to mechanically movable blades [23]. The velocities corresponding to Equations (1) and (2) are

$$\begin{aligned}\dot{x} &= v_x\cos\psi\cos\theta + v_y(-\sin\psi\cos\phi + \cos\psi\sin\theta\\ &\quad \sin\phi) + v_z(\sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi)\\ \dot{y} &= v_x\sin\psi\cos\theta + v_y(\cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi)\\ &\quad + v_z(-\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi)\\ \dot{z} &= v_x\sin\theta - v_y\cos\theta\sin\phi - v_z\cos\theta\cos\phi\end{aligned} \tag{5}$$

and

$$\begin{aligned}\dot{\theta} &= w_y\cos\phi - w_z\sin\phi\\ \dot{\phi} &= w_x + w_y\sin\phi\tan\theta + w_z\cos\phi\tan\theta\\ \dot{\psi} &= w_y\frac{\sin\phi}{\cos\theta} + w_z\frac{\cos\phi}{\cos\theta}\end{aligned} \tag{6}$$

respectively. The Equations (1)–(6) represent the complete nonlinear model of a quadcopter, composed of 12 states, 4 inputs, and 12 outputs. More precisely,

$$\mathbf{x} = \begin{bmatrix} v_x & v_y & v_z & w_x & w_y & w_z & \theta & \phi & \psi & x & y & z \end{bmatrix}^{\mathrm{T}} \tag{7}$$

is the state or system vector,

$$\mathbf{u} = \begin{bmatrix} T & M_x & M_y & M_z \end{bmatrix}^{\mathrm{T}} \tag{8}$$

is the input or control vector,

$$\mathbf{y} = \mathbf{x} \tag{9}$$

is the output (measured) vector. Furthermore, the reduced state vector

$$\mathbf{x_s} = \begin{bmatrix} v_x & v_y & v_z & w_x & w_y & w_z \end{bmatrix}^{\mathrm{T}} \tag{10}$$

and the performance outputs

$$y_{p_1} \in \{x, y, z\}, \tag{11}$$

$$y_{p_2} \in \{\theta, \phi, \psi\}, \tag{12}$$

and

$$\mathbf{y_p} = \begin{bmatrix} x & y & z \end{bmatrix}^{\mathrm{T}} \tag{13}$$

are defined for future use.

Using standard linearization, nonlinear dynamic equations can be converted into a set of standard linear equations to examine the stability and controllability of the system as well as to design an LQR integral control to compare with SMC PD method. This yields,

$$\begin{aligned} \dot{\mathbf{x}} &= \left[-g\theta \;\; g\phi \;\; -\frac{T}{m} \;\; \frac{M_x}{J_x} \;\; \frac{M_y}{J_y} \;\; \frac{M_z}{J_z} \;\; w_y \;\; w_x \;\; w_z \;\; v_x \;\; v_y \;\; -v_z \right]^{\mathrm{T}} \\ \mathbf{y} &= \mathbf{x} \end{aligned} \tag{14}$$

that can further written into the standard state space form

$$\begin{aligned} \dot{\mathbf{x}} &= A\mathbf{x} + B\mathbf{u} \\ \mathbf{y} &= C\mathbf{x} + D\mathbf{u} \end{aligned} \tag{15}$$

where A, B, C, and D are known as the state or system matrix, input or control matrix, output (measured) matrix, and feedthrough matrix respectively. Correspondingly, **x**, **u**, and **y** are known as the state or system vector, input or control vector, and output (measured) vector as in Equations (7)–(9). The system parameters are taken from [20] and illustrated in Table 1.

TABLE 1. System parameters.

Symbol	Quantity	Value
g	gravitational force	9.81 m/s^2
ℓ	length of the fixed pitch to mechanically movable blades	0.2 m
m	mass of quadcopter	0.8 kg
J_{mp}	moment of inertia of motor with propeller	≈ 0
J_x, J_y	moment of inertia w.r.t. axis x, y	1.8×10^{-3} kgm^2
J_x	moment of inertia w.r.t. axis z	1.5×10^{-3} kgm^2
k_{MT}	ratio of the reactive moment and thrust	0.1 m

Refer to linear model Equation (15) and system parameters in Table 1, the system has 12 eigenvalues at the origin. Prior to any control design, the controllability matrix $\begin{bmatrix} B & AB & A^2B & \ldots & A^{n-1}B \end{bmatrix}$ is calculated and found to be of rank 12. Thus, all the 12 states are controllable.

IV. CONTROL DESIGNS FOR THE SWARM OF DRONES

A hierarchical swarm of drones consisting of ten quadcopters is considered in a leader–follower operational configuration, as presented in Fig. 2. The states/nodes can be seen to form clusters at each level of the hierarchy. The distance between any two neighbouring nodes p and q is defined as $d_{p,q}$ where p, q $\in$ {L, 1–9}, and each node is operated on its internal control system. The straight arrows show the direction in which coordinate variables are shared.

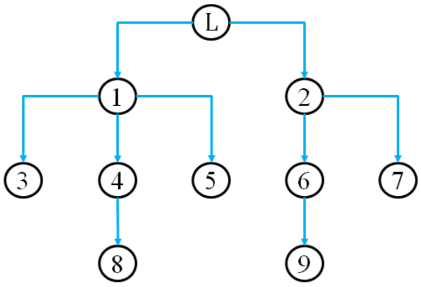


FIGURE 2. Organization of the considered swarm of drones.

The swarm in Fig. 2 is composed of the main leader L and multiple layers of sub-leaders forming interconnected sub-clusters/swarms of different sizes. At any level of the hierarchy, each drone is directly communicating with its respective neighbours, i.e. with its immediate leader and follower, by providing its orientation information that results in maintaining the distance between any two adjacent nodes in the swarm [1]. The control architecture, in a swarm of drones, is responsible for tracking the generated reference commands in a stabilized manner while keeping a safe distance between neighbouring drones to maintain the desired flight formation. To form the swarm based on the partially distributed controller on each node in such a system, the leader receives known bounded reference commands,

$$r_1 \in \{x_r, y_r, z_r\} \tag{16}$$

and

$$\mathbf{r} = \begin{bmatrix} x_r & y_r & z_r \end{bmatrix}^{\mathrm{T}}, \tag{17}$$

that may or may not vary with time and the controller of the follower node is dependent on its respective leader's orientation and movement. The translational positions x_i, y_i and z_i, $i \in$ {L, 1–9}, are fed to the respective follower(s) as a reference. On top of that, all followers have their local offsets in order to avoid collisions with each other.

Due to the hierarchical nature of the architecture as a whole, the main problem in the adjustment of the controller's design parameters is the dependence of each drone's location on the other drones' locations in the swarm. In other words, deviations in the control design parameters of a drone might cause significant unwanted changes in the other drones' locations, especially in the case of the follower nodes, which can be observed as oscillations. The parameters of the controller are determined through testing of the overall dynamics of the swarm. The goal of the control system is to improve the tracking performance in such a way that the system's settling time is reduced as much as possible without introducing oscillations in the response. To accomplish this, a nonlinear SMC PD and a linear LQR integral action control design methods are proposed.

A. SMC PD CONTROL SYSTEM

An SMC PD control scheme from [24] and [25] has been adapted for this study and tailored for the leader–follower tightly coupled formation flight. This technique has been selected because of its high flexibility and reconfigurability for different formations. Another advantage of using SMC is that the system is insensitive to disturbances when it enters the sliding surface. However, it leads to a chattering phenomenon that is reduced by using proper switching gains in the saturation function. Fig. 3 shows the decision-making process of each drone that is split into two feedback control blocks, i.e outer and inner feedback loops.

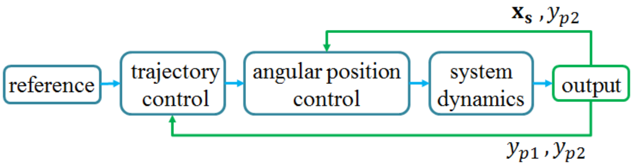


FIGURE 3. Block diagram of SMC PD control design.

In the control design depicted in Fig. 3, SMC is tied to the PD structure and the parameters can be adjusted based on the dynamics of the swarm. The outer feedback loop, section IV-A1, is responsible for the drone's trajectory, i.e. translational positions x, y, and z, whereas the inner feedback loop, section IV-A2, is manipulating the control variables of actuator thrust T and the torques M_i.

1) TRAJECTORY CONTROL DESIGN

The internal structure of each drone's trajectory tracking control block is illustrated in Fig. 4. The angular errors, $e_2 \in \{\theta_e, \phi_e, \psi_e\}$, are generated using three PD controllers that are fed as inputs to the angular position stabilization block.

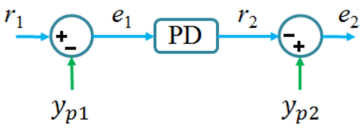


FIGURE 4. The trajectory control structure.

From Fig. 4, the feedback law

$$e_2 = y_{p_2} - (K_P \cdot (r_1 - y_{p_1}) + K_D \frac{d}{dt}(r_1 - y_{p_1})) \quad (18)$$

is obtained. The signals y_{p1} and y_{p2} are given in Equations (11) and (12) respectively. The parameters K_P and K_D are the proportional and derivative gains respectively of the PD controller.

To reduce the overshoots of translational trajectories x, y, and z, a lead compensator is connected in series with the PD control block. This compensation is included in level 2 (Fig. 2, nodes 1 and 2) for y position, and in levels 3–4 (Fig. 2, nodes 3–9) for x and y positions. Mathematically, the transfer function

$$G(s) = \frac{s - z_z}{s - z_p} \quad (19)$$

of the lead compensator, where z_z is the zero, and z_p is the pole satisfying $0 < z_z < z_p$. The parameters used in this study are given in Appendix A.

2) ANGULAR POSITION CONTROL DESIGN

SMC control design is used for generating the actuator thrust T, to drive each drone in a swarm formation at the desired altitude, and the torques M_i, for the stabilization of the orientation angles, as pictured in Fig. 5. This practice reduces the order of the state equations and provides a quick response.

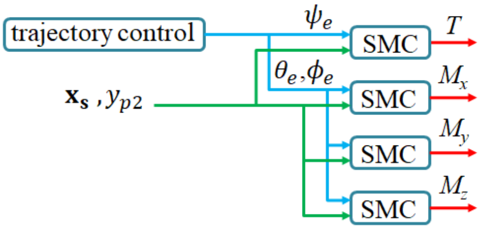


FIGURE 5. The angular position control structure.

Consider a time varying surface $s_a(t)$ in the state space $\mathbb{R}^n$ by the scalar equation $s_a(x, t) = 0$. The sliding surface is chosen as

$$s_\alpha = c_\alpha e_\alpha + \dot{e}_\alpha \quad (20)$$

where $\alpha \in \{T, M_x, M_y, M_z\}$, e_α is the tracking error, and c_α is a strictly positive constant [26]. The four control laws have been proposed here for thrust T and torques M_i; M_x (along x-axis, i.e. for roll ϕ), M_y (along y-axis, i.e. for pitch θ), and M_z (along z-axis, i.e. for yaw ψ). The chattering free control law is given by

$$u_\alpha = \hat{u}_\alpha - K_\alpha \text{sat}(\frac{s_\alpha}{\beta_\alpha}) \quad (21)$$

where $\hat{u}_\alpha$ and K_α are the control design parameters. The $\hat{u}_\alpha$ indicates the motion of the state trajectory along the sliding surface s_α, and K_α represents the maximum controller output. The sat$(\cdot)$ is the saturation function defined as

$$\text{sat}(\frac{s_\alpha}{\beta_\alpha}) = \begin{cases} \frac{s_\alpha}{\beta_\alpha}, & |\frac{s_\alpha}{\beta_\alpha}| \le 1 \\ \text{sgn}(\frac{s_\alpha}{\beta_\alpha}), & \text{otherwise} \end{cases} \quad (22)$$

where β_α is a constant that defines the thickness of the boundary layer. The saturation function is used to smooth out the control discontinuity in a thin boundary layer neighbouring the switching surfaces. To accomplish a trade-off between tracking precision and robustness to unmodeled dynamics, this function allocates a low pass filter to the local dynamics of the variable s_α, thus eliminating chattering phenomena. The control laws are much more efficient when the minimum amount of boundary layer is introduced. The equivalent control $\hat{u}_\alpha$,

$$\begin{aligned}\hat{u}_T &= -\frac{mc_T}{\cos\theta\cos\phi}(v_x\sin\theta - v_y\cos\theta\sin\phi - v_z\\ &\quad\cos\theta\cos\phi) + m(-v_y w_x + v_x w_y\\ &\quad + g\cos\theta\cos\phi)\\ \hat{u}_{M_x} &= -c_x J_x(w_x + w_y\sin\phi\tan\theta + w_z\cos\phi\tan\theta)\\ &\quad + w_y w_z(J_z - J_y)\\ \hat{u}_{M_y} &= -\frac{c_y J_y}{\cos\phi}(w_y\cos\phi - w_z\sin\phi) + w_x w_z\\ &\quad (J_x - J_z)\\ \hat{u}_{M_z} &= -c_z J_z(w_y\tan\phi + w_z),\end{aligned}\tag{23}$$

is obtained using nonlinear model Equations (1)–(6). The control law presented in Equation (21) ensure both the reachability condition, $s_\alpha\dot{s}_\alpha < 0$, and the sliding condition, $\dot{V}_\alpha = \frac{1}{2}\frac{d}{dt}s_\alpha^2$, using Lyapunov's stability analysis that is described below.

Consider a positive definite scalar function for the thrust T and torques M_i as

$$V_\alpha = \frac{1}{2}s_\alpha^2 \tag{24}$$

and its derivative leads to

$$\dot{V}_\alpha = s_\alpha\dot{s}_\alpha \tag{25}$$

where $\dot{s}_\alpha$,

$$\begin{aligned}\dot{s}_T &= c_T(v_x\sin\theta - v_y\cos\theta\sin\phi - v_z\cos\theta\cos\phi)\\ &\quad - (-v_y w_x + v_x w_y + g\cos\theta\cos\phi - \frac{u_T}{m})\\ &\quad \cos\theta\cos\phi\\ \dot{s}_{M_x} &= c_x(w_x + w_y\sin\phi\tan\theta + w_z\cos\phi\tan\theta)\\ &\quad + \frac{1}{J_x}(-w_y w_z(J_z - J_y) + u_{M_x})\\ \dot{s}_{M_y} &= c_y(w_y\cos\phi - w_z\sin\phi) + \frac{\cos\phi}{J_y}(-w_x w_z\\ &\quad (J_x - J_z) + u_{M_y})\\ \dot{s}_{M_z} &= c_z(w_y\frac{\sin\phi}{\cos\theta} + w_z\frac{\cos\phi}{\cos\theta}) + \frac{u_{M_z}}{J_z}\frac{\cos\phi}{\cos\theta},\end{aligned}\tag{26}$$

is obtained using nonlinear model Equations (1)–(6). Keeping the control law from Equation (21) in Equation (25),

$$\dot{V} = -\eta|s_\alpha| < 0 \tag{27}$$

is obtained where $\eta = 1$, a strictly positive constant. It is proved from Equation (27) that all the system trajectories point towards the sliding surface s_α in a finite time. To be specific, once on the surface, the system trajectories stay on the surface. In a nutshell, sliding condition makes the surface an invariant set. Some dynamic disturbances can be tolerated while still keeping the surface an invariant set. The conditions are verified by s_α and therefore the inner closed-loop system is guaranteed to be stable. The outer PD loop is not taken into account in the analysis, and it can affect the stability of the whole system. However, a PD loop has good stability properties, and SMC cancels to some extent the nonlinearity of the system, so it is unlikely that the PD will cause instability. The parameters used in this study are given in Appendix A.

B. LQR INTEGRAL ACTION CONTROL SYSTEM

LQR is a simple control design method that can handle a multivariable system and is known to provide good robustness properties in the full state feedback case [27]. Therefore, a standard LQR augmented with integral action is tested, similar to what was done in [28] and [29], for the leader–follower tightly coupled formation flight. Each drone has its control system, with the setpoints coming from the above level. Based on linear model Equation (15), LQR provides an optimal full state feedback controller that is applied on the nonlinear model Equations (1)–(6) for each quadcopter in the swarm under consideration.

Considering the linear model Equation (15), the control input $\mathbf{u}$ minimizes the quadratic cost function

$$J(\mathbf{u}) = \int_0^\infty(\dot{\mathbf{x}}_\mathbf{a}^\mathrm{T} Q\dot{\mathbf{x}}_\mathbf{a} + \dot{\mathbf{u}}^\mathrm{T} R\dot{\mathbf{u}})dt \tag{28}$$

where

$$\mathbf{x_a} = \left[\int_0^t \mathbf{e}(\tau)^\mathrm{T}\,d\tau \quad \mathbf{x}^\mathrm{T}\right]^\mathrm{T}, \tag{29}$$

the states $\mathbf{x}$ and the control vector $\mathbf{u}$ are defined in Equations (7) and (8) respectively. Q is a positive semi-definite matrix that defines the weights on states, whereas R is a positive definite matrix that weights the control inputs. Furthermore,

$$\mathbf{e} = \mathbf{r} - \mathbf{y_p} \tag{30}$$

is the error term, and $\mathbf{r}$ from Equation (17) is the reference for $\mathbf{y_p}$ that is defined in Equation (13). To get the desired response, the controller can be tuned by changing the (diagonal) entries in the Q and R matrices. The state feedback law

$$\mathbf{u} = \frac{K_i}{s}\mathbf{e} - K_p\mathbf{x} \tag{31}$$

gives the four control inputs, i.e. thrust T and torques M_i; M_x (along x-axis, i.e. for roll ϕ), M_y (along y-axis, i.e. for pitch θ), and M_z (along z-axis, i.e. for yaw ψ). The controller uses two separate gain matrices K_p and K_i where K_p is the state feedback gain, and K_i is the integral gain. The feedback law is shown in Figure 6.

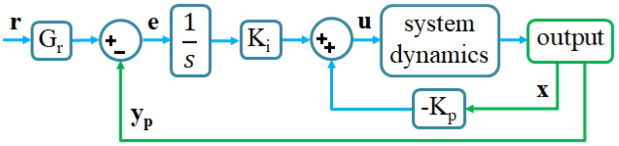


FIGURE 6. Block diagram of LQR integral control design.

The tuning parameters Q and R used in this study are given in Appendix B.

V. SIMULATION SETUP

The system architecture of the swarm based on a leader–follower hierarchical formation that consists of four levels of hierarchy in a rigid environment, as shown in Fig. 2, is simulated in Simulink® MATLAB. Level one consists of the leader L, level two of the followers 1–2, level three of the followers 3–7, and level four of the followers 8–9. The control systems of each drone follow the reference commands that are defined by the pre-specified formation and its respective leader. In the design of the MISO nonlinear SMC PD control technique, different gain adjustments are performed to reconfigure the controller based on the dynamics of the fleet whereas in the design of the MIMO optimal control using LQR integral action, the same weights are used in the controllers of each drone. The initial launching position x of each drone is set to 1m away from its respective neighbouring drones, and the further data for simulation is shown in Table 2. In all simulations, the sampling time of 0.01s is used for all the figures.

TABLE 2. Initial positions (m) and offsets (m) of drones used in simulation.

Drones	Symbol	Initial Position $(x, y, z)^*$	Offset $(x, y, z)^*$
Leader	L	$(0, 0, 0)$	none
Follower 1	f1	$(1, 0, 0)$	$(1, 0, 0)$
Follower 2	f2	$(-1, 0, 0)$	$(-1, 0, 0)$
Follower 3	f3	$(2, 0, 0)$	$(1, 0, 0)$
Follower 4	f4	$(3, 0, 0)$	$(2, 0, 0)$
Follower 5	f5	$(4, 0, 0)$	$(3, 0, 0)$
Follower 6	f6	$(-2, 0, 0)$	$(-1, 0, 0)$
Follower 7	f7	$(-3, 0, 0)$	$(-2, 0, 0)$
Follower 8	f8	$(5, 0, 0)$	$(2, 0, 0)$
Follower 9	f9	$(-4, 0, 0)$	$(-2, 0, 0)$

Three different scenarios are used to evaluate the proposed distributed controller in case of similar movement of the swarm of drones. The reference commands, specified in Table 3, are given to the leader drone, and all the followers track its position with safe distance strategy using offsets, stated in Table 2, in order to avoid collisions with each other. In the first scenario, the tracking performances of the swarming drones is examined using a step change reference, see Fig. 7(a). In the second scenario, a circular reference based on lemniscate of gerono equations, also known as 8 curve, is used, and the results are displayed in Fig. 7(b). In the third scenario, another circular reference based on Archimedean spiral equations, also known as an arithmetic spiral, is tracked in Fig. 7(c).

VI. RESULTS

The 3D response of the formation for all three reference models, as presented in Fig. 7, depicts that the leader drone tracks the reference signal and the followers track the output of the associated leader in steady hierarchical manner while keeping the desired distance with its neighbouring peer(s), i.e. collision avoidance using offsets mentioned in Table 2. The corresponding trajectory errors w.r.t. related reference trajectories using both control designs, depicted in Fig. 8, show the average of the error data points for the trajectories of the swarm which is referred to as arithmetic mean. It is understandable from the graphs that the performance of the SMC PD control design is seen better in comparison to the LQR integral action control strategy in most of the drones due to its low average error.

Only the results for the step change reference model are illustrated in Fig. 9 for the sake of brevity. Fig. 9(a)–(e) describe the control actions, thrust T and torques M_i, of drones in order to lift, fly, and track the given trajectory. The performance outputs x, y, and z, in Fig. 9(f)–(h), have rapid response and settling time w.r.t. their setpoints using the SMC PD control design in contrast to the LQR integral action strategy. Also in Fig. 9(i)–(l), the sliding surfaces of the SMC PD control design converge to zero in a finite time by driving the respective system states to reach their sliding surfaces.

For all reference trajectories, the total kinetic energy KE produced by the swarm due to its motion versus the total stored potential energy PE is pictured in Fig. 10. Total energy possessed and held is calculated as $KE = 0.5mv^2$ and $PE = mgh$ respectively. These energies relate how much work is conserved in the process of the swarm movement. Each drone in the swarm depends fully on the variable speed of the motors driving the propellers. This variation, along with the rpm thrust and control in propeller/motor speed, provides the quadcopter with all of the essential control to fly. Flying with payload affects the control and the stability of a drone. The performance of the swarm in terms of work done and energy stored w.r.t. time is calculated by

$$E_{KE} = \frac{KE_{\text{max, LQR int}} - KE_{\text{max, SMC PD}}}{KE_{\text{max, LQR int}} + KE_{\text{max, SMC PD}}} \times 100\% \qquad (32)$$

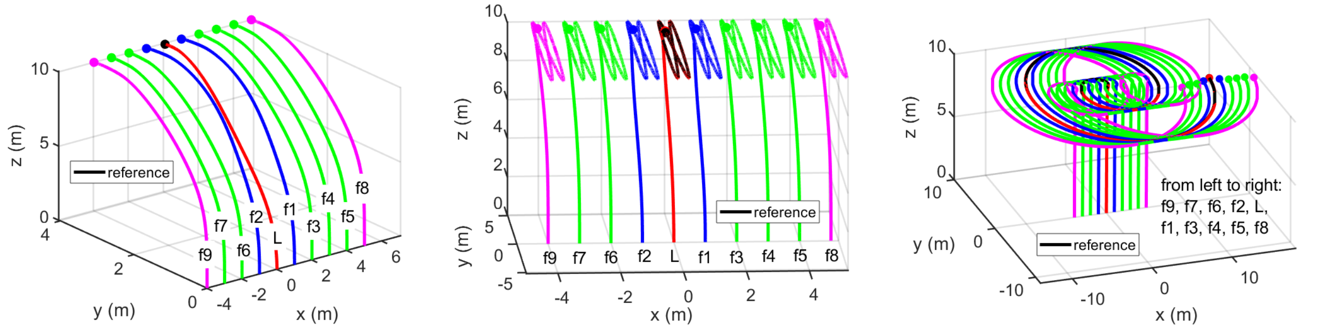
and

$$E_{PE} = \frac{t_{\text{PE, LQR int}} - t_{\text{PE, SMC PD}}}{t_{\text{PE, LQR int}}} \times 100\% \qquad (33)$$

respectively, where E_{KE} and E_{PE} are defined as performance indices of kinetic and potential energies respectively. The KE_{max} is represented as the peak value of the kinetic energy

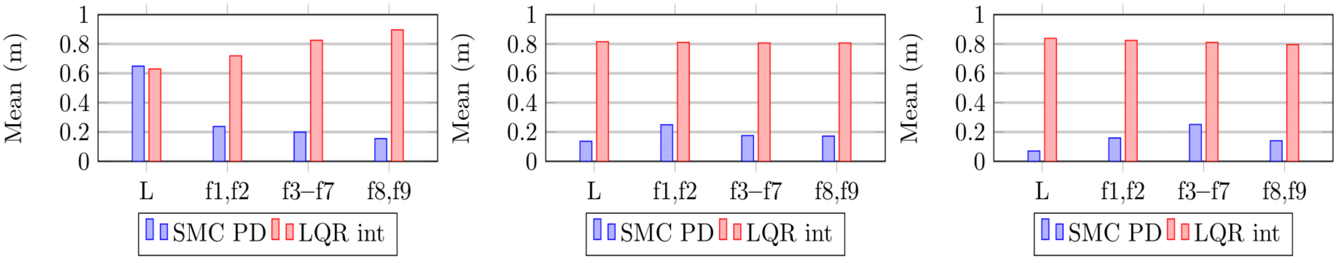
TABLE 3. Test sequences for simulations. The $\mathcal{T}$ is defined as a ramp function with the slope of 1.

Commanded Positions for the Leader	x (m)	y (m)	z (m)	Simulation's Running Time, t (s)
Step change at time $t = 0$s	$0 \longrightarrow 2$	$0 \longrightarrow 4$	$0 \longrightarrow 10$	25
8 curve	$-0.25\cos 2\pi 0.05\mathcal{T}$	$10\sin 2\pi 0.05\mathcal{T}\cos 2\pi 0.05\mathcal{T}$	$0 \longrightarrow 10$	150
Arithmetic spiral	$0.05\mathcal{T}\cos 2\pi 0.05\mathcal{T}$	$0.05\mathcal{T}\sin 2\pi 0.05\mathcal{T}$	$0 \longrightarrow 10$	250



(a) With step change (b) With 8 curve (c) With arithmetic spiral

FIGURE 7. Tracking performance of swarming drones, using SMC PD, for the reference models.



(a) With step change (b) With 8 curve (c) With arithmetic spiral

FIGURE 8. Mean value of Euclidean distance from setpoints for the two different control designs.

TABLE 4. Overall system performance indices, defined in Equations (32) and (33).

Reference Models	$KE_{\text{max, LQR int}}$ (J)	$KE_{\text{max, SMC PD}}$ (J)	E_{KE} (%)	$t_{\text{PE, LQR int}}$ (s)	$t_{\text{PE, SMC PD}}$ (s)	E_{PE} (%)
Step change	40.3168	54.0619	−14.5638	7.6156	4.0202	47.211
8 curve	39.2656	55.4020	−17.0453	7.6346	4.4042	42.3126
Arithmetic spiral	39.2149	55.5315	−17.2213	7.6321	4.3616	42.8519

whereas t_{PE} is the settling time of the potential energy. All the obtained values, using MATLAB command *stepinfo*, are given in Table 4. For the reference commands, the corresponding settling time of the swarm is improved by 47.2%, 42.3%, and 42.8% respectively using the SMC PD control design in comparison to the LQR integral action strategy. As a consequence of this, the position and velocity errors converge much quickly and smoothly. Hence, the work is performed faster as the energy is transmitted or used in lesser time.

The distance between any two neighbouring nodes p and q in Fig. 2 is calculated by

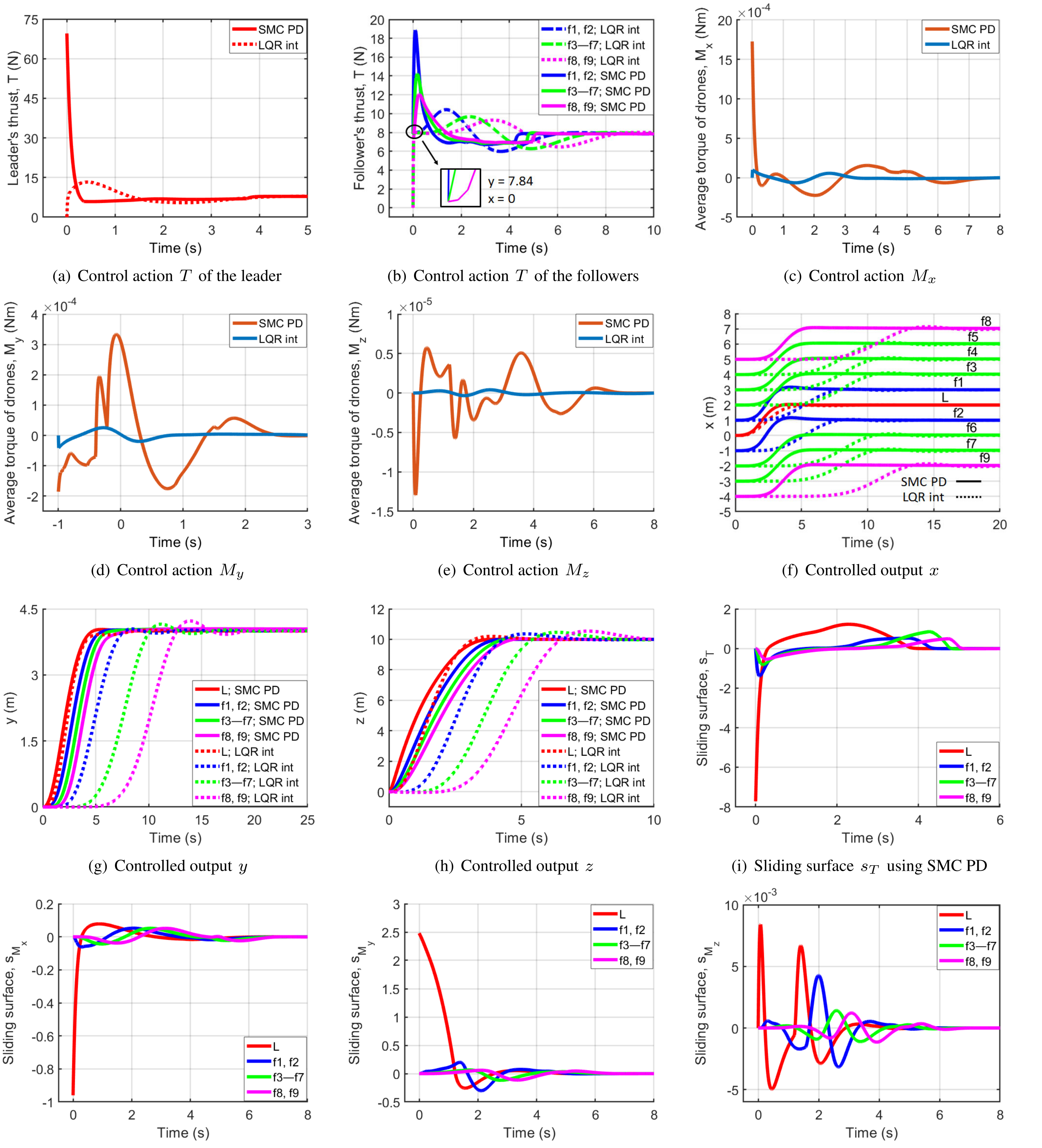
$$d_{p,q} = d_{q,p} = \sqrt{(q_x - p_x)^2 + (q_y - p_y)^2 + (q_z - p_z)^2} \quad (34)$$

where its minimum distance d_{min} should be greater than zero in order to hold the collision avoidance phenomenon. The results of nominal and minimum distances, d_{nom} and d_{min} respectively, are presented in Tables 5–7. It is evident from the results that both techniques fulfil the collision avoidance constraint by maintaining the desired separations from the respective nodes. There is a lower risk of drone's collisions using the SMC PD control design for all the reference commands. On the other hand there is a high risk of drone's collisions using distance separation of 1m between the respective neighbours with the LQR integral action control strategy for spiral reference model. Therefore, in this case, it is recommended to set the offset greater than 2m between any two neighbouring nodes in order to avoid collisions with each other.

Using derivatives of Equation (28), the cost computations that are minimized by control inputs **u** are given in Table 8. From the cost computations of proposed controllers, it is asserted that the overall hierarchical swarm has a lower total cost with the SMC PD control design, i.e. 2.73E+07, 1.52E+08, and 2.29E+08 for the corresponding references whereas with the LQR integral action control strategy, the corresponding total computations are 3.29E+07, 1.58E+08, and 2.58E+08.

A. PLANT-MODEL MISMATCH

In this section, the swarm of drones is implemented with a plant-model mismatch to observe the impact on the proposed control designs in the hierarchical levels of the swarm for the parametric and input uncertainties. For better clarity, the simulation is only conducted for the step change reference commands.



(a) Control action T of the leader

(b) Control action T of the followers

(c) Control action M_x

(d) Control action M_y

(e) Control action M_z

(f) Controlled output x

(g) Controlled output y

(h) Controlled output z

(i) Sliding surface s_T using SMC PD

(j) Sliding surface s_{M_x} using SMC PD

(k) Sliding surface s_{M_y} using SMC PD

(l) Sliding surface s_{M_z} using SMC PD

FIGURE 9. Results with the step change reference model.

In the case of parametric uncertainties, the mass m,

$$\begin{aligned} m_{\mathrm{L}} &= 0.9\mathrm{kg} \\ m_{\mathrm{f1,f2}} &= 1\mathrm{kg} \\ m_{\mathrm{f3\text{–}f7}} &= 1.1\mathrm{kg} \\ m_{\mathrm{f8,f9}} &= 1.2\mathrm{kg}, \end{aligned} \tag{35}$$

is set for the quadcopter(s) on each level of the swarm. This is to be considered here that for the SMC PD control design, the mass should be greater than 0.8kg for the stability of its altitude. In the case of input uncertainties, the closed-loop system of each quadcopter has different thrust input $T_{real} = T \times K$, where T is the output of the controllers and

TABLE 5. The nominal and minimum distances (m) between neighbouring drone(s) in the swarm, with step change.

	from / to	f1	f2	f3	f4	f5	f6	f7	f8	f9
L	d_{nom}	1	1	2	3	4	2	3	5	4
	$d_{\text{min, LQR int}}$	0.8033	0.9323	1.6362	2.5173	3.4096	1.9043	2.9029	4.2580	3.8622
	$d_{\text{min, SMC PD}}$	1	0.9064	1.8941	2.7478	3.6130	1.9364	2.9364	4.5601	3.9191
f1	d_{nom}		2	1	2	3	3	4	4	5
	$d_{\text{min, LQR int}}$		2	0.7332	1.5957	2.4819	2.8663	3.8650	3.3499	4.8729
	$d_{\text{min, SMC PD}}$		2	0.7166	1.4125	2.2668	2.9674	3.9674	3.2265	4.9594
f2	d_{nom}			3	4	5	1	2	6	3
	$d_{\text{min, LQR int}}$			2.4819	3.3851	4.3026	0.8764	1.8688	5.1481	2.8751
	$d_{\text{min, SMC PD}}$			2.2668	3.2025	4.1676	0.9675	1.9675	5.0283	2.9594
f3	d_{nom}				1	2	4	5	3	6
	$d_{\text{min, LQR int}}$				1	2	4	5	2.4478	5.8237
	$d_{\text{min, SMC PD}}$				1	2	4	5	2.7495	5.9773
f4	d_{nom}					1	5	6	2	7
	$d_{\text{min, LQR int}}$					1	5	6	1.5061	6.7514
	$d_{\text{min, SMC PD}}$					1	5	6	1.5527	6.8231
f5	d_{nom}						6	7	1	8
	$d_{\text{min, LQR int}}$						6	7	0.6824	7.8226
	$d_{\text{min, SMC PD}}$						6	7	0.9921	7.9768
f6	d_{nom}							1	7	2
	$d_{\text{min, LQR int}}$							1	6.1784	1.8329
	$d_{\text{min, SMC PD}}$							1	6.6736	1.9808
f7	d_{nom}								8	1
	$d_{\text{min, LQR int}}$								7.1379	0.8460
	$d_{\text{min, SMC PD}}$								7.6668	0.9853
f8	d_{nom}									9
	$d_{\text{min, LQR int}}$									9
	$d_{\text{min, SMC PD}}$									9

TABLE 6. The nominal and minimum distances (m) between neighbouring drone(s) in the swarm, with 8 curve.

	from / to	f1	f2	f3	f4	f5	f6	f7	f8	f9
L	d_{nom}	1	1	2	3	4	2	3	5	4
	$d_{\text{min, LQR int}}$	0.9770	0.9770	1.9539	2.9539	3.9538	1.9539	2.9539	4.9307	3.9309
	$d_{\text{min, SMC PD}}$	0.9992	0.9995	1.9721	2.9721	3.9721	1.9979	2.9979	4.9644	3.9964
f1	d_{nom}		2	1	2	3	3	4	4	5
	$d_{\text{min, LQR int}}$		2	0.9768	1.9768	2.9768	2.8908	3.8839	3.9536	4.8512
	$d_{\text{min, SMC PD}}$		2	0.9728	1.9728	2.9728	2.9979	3.9807	3.9650	4.9964
f2	d_{nom}			3	4	5	1	2	6	3
	$d_{\text{min, LQR int}}$			2.9768	3.9767	4.9767	0.9440	1.9041	5.9535	2.8879
	$d_{\text{min, SMC PD}}$			2.9728	3.9728	4.9728	0.9981	1.9980	5.9650	2.9965
f3	d_{nom}				1	2	4	5	3	6
	$d_{\text{min, LQR int}}$				1	2	4	5	2.9766	5.8932
	$d_{\text{min, SMC PD}}$				1	2	4	5	2.9922	5.9938
f4	d_{nom}					1	5	6	2	7
	$d_{\text{min, LQR int}}$					1	5	6	1.9767	6.8884
	$d_{\text{min, SMC PD}}$					1	5	6	1.9922	6.8884
f5	d_{nom}						6	7	1	8
	$d_{\text{min, LQR int}}$						6	7	0.9767	7.8847
	$d_{\text{min, SMC PD}}$						6	7	0.9922	7.9841
f6	d_{nom}							1	7	2
	$d_{\text{min, LQR int}}$							1	6.9765	1.9416
	$d_{\text{min, SMC PD}}$							1	6.9922	1.9985
f7	d_{nom}								8	1
	$d_{\text{min, LQR int}}$								7.9764	0.9483
	$d_{\text{min, SMC PD}}$								7.9922	0.9985
f8	d_{nom}									9
	$d_{\text{min, LQR int}}$									9
	$d_{\text{min, SMC PD}}$									9

TABLE 7. The nominal and minimum distances (m) between neighbouring drone(s) in the swarm, with arithmetic spiral.

	from / to	f1	f2	f3	f4	f5	f6	f7	f8	f9
L	d_{nom}	1	1	2	3	4	2	3	5	4
	$d_{\text{min, LQR int}}$	0.3505	0.0347	0.7046	0.1401	1.1364	0.0639	0.9294	0.7244	0.9089
	$d_{\text{min, SMC PD}}$	0.9885	0.9908	1.4463	2.4462	3.4462	1.6037	2.6035	4.2635	3.4729
f1	d_{nom}		2	1	2	3	3	4	4	5
	$d_{\text{min, LQR int}}$		2	0.3510	0.5655	1.5636	1.9594	2.9586	1.1379	2.9251
	$d_{\text{min, SMC PD}}$		2	0.4545	1.4542	2.4541	2.6091	3.6091	3.2713	4.4783
f2	d_{nom}			3	4	5	1	2	6	3
	$d_{\text{min, LQR int}}$			1.5636	2.5627	3.5622	0.0337	0.9611	3.1339	0.9307
	$d_{\text{min, SMC PD}}$			2.4541	3.4541	4.4541	0.6094	1.6092	5.2713	2.4785
f3	d_{nom}				1	2	4	5	3	6
	$d_{\text{min, LQR int}}$				1	2	4	5	1.5641	4.9580
	$d_{\text{min, SMC PD}}$				1	2	4	5	2.8172	5.8692
f4	d_{nom}					1	5	6	2	7
	$d_{\text{min, LQR int}}$					1	5	6	0.5663	5.9578
	$d_{\text{min, SMC PD}}$					1	5	6	1.8172	6.8692
f5	d_{nom}						6	7	1	8
	$d_{\text{min, LQR int}}$						6	7	0.3516	6.9576
	$d_{\text{min, SMC PD}}$						6	7	0.8172	7.8692
f6	d_{nom}							1	7	2
	$d_{\text{min, LQR int}}$							1	5.5618	0.9616
	$d_{\text{min, SMC PD}}$							1	6.8172	1.8692
f7	d_{nom}								8	1
	$d_{\text{min, LQR int}}$								6.5616	0.0327
	$d_{\text{min, SMC PD}}$								7.8172	0.8693
f8	d_{nom}									9
	$d_{\text{min, LQR int}}$									9
	$d_{\text{min, SMC PD}}$									9

TABLE 8. LQR cost, $\int \dot{\mathbf{e}}^{\mathrm{T}} Q_e \dot{\mathbf{e}}\, dt + \int \dot{\mathbf{x}}^{\mathrm{T}} Q_x \dot{\mathbf{x}}\, dt + \int \dot{\mathbf{u}}^{\mathrm{T}} R \dot{\mathbf{u}}\, dt$, where Q_e and Q_x are diagonal matrices with corresponding weights from Q.

LQR cost	L			f1, f2			f3–f7			f8, f9		
	step	8 curve	spiral	step	8 curve	spiral	step	8 curve	spiral	step	8 curve	spiral
from e	1.37E+06	1.36E+06	1.37E+06	1.06E+06	1.05E+06	1.05E+06	9.23E+05	9.19E+05	9.20E+05	8.44E+05	8.42E+05	8.44E+05
from e for SMC PD	2.27E+06	2.24E+06	2.24E+06	1.36E+05	1.42E+05	1.42E+05	5.74E+04	5.87E+04	5.91E+04	4.35E+04	4.35E+04	4.36E+04
from $\dot{x}$	2.55E+06	1.50E+07	2.51E+07	2.41E+06	1.49E+07	2.49E+07	2.28E+06	1.48E+07	2.48E+07	2.17E+06	1.47E+07	2.47E+07
from $\dot{x}$ for SMC PD	2.55E+06	1.51E+07	1.42E+06	2.47E+06	1.50E+07	2.50E+07	2.43E+06	1.49E+07	2.49E+07	2.39E+06	1.49E+07	2.49E+07
from $\dot{u}$	1.57E+03	9.27E+03	1.54E+04	1.55E+03	9.25E+03	1.54E+04	1.55E+03	9.25E+03	1.54E+04	1.55E+03	9.25E+03	1.54E+04
from $\dot{u}$ for SMC PD	1.68E+03	9.38E+03	1.55E+04	1.57E+03	9.27E+03	1.54E+04	1.55E+03	9.26E+03	1.54E+04	1.55E+03	9.26E+03	1.54E+04
Total	3.92E+06	1.64E+07	2.64E+07	3.46E+06	1.60E+07	2.60E+07	3.21E+06	1.57E+07	2.57E+07	3.02E+06	1.55E+07	2.55E+07
Total for SMC PD	4.82E+06	1.73E+07	3.67E+06	2.61E+06	1.51E+07	2.51E+07	2.48E+06	1.50E+07	2.50E+07	2.43E+06	1.49E+07	2.49E+07

T_{real} is the actual thrust obtained, and the gain K,

$$\begin{aligned} K_L &= 0.9 \\ K_{\text{f1,f2}} &= 0.8 \\ K_{\text{f3–f7}} &= 0.7 \\ K_{\text{f8,f9}} &= 0.6, \end{aligned} \tag{36}$$

is set for the quadcopter(s) on each level of the swarm. It is to be noted here that for the SMC PD control design, the gain should be less than 1 for the stability of its altitude. Rest of the data for both tests is given in Tables 1–3, and the acquired results are shown in Fig. 11 and Fig. 12. It is presented, in Fig. 11(c) and Fig. 12(c) that when using the SMC PD control design, the z positions have undershoots in all the levels of hierarchical formation compared to its nominal case. Also in the altitudes, the acceptable tracking errors w.r.t. their settling points exist and may increase with the increase in the mass of the quadcopter(s) whereas, the LQR integral action control strategy is very much insensitive to such disturbances.

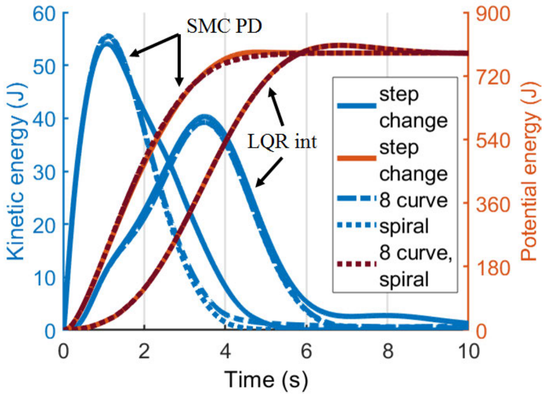
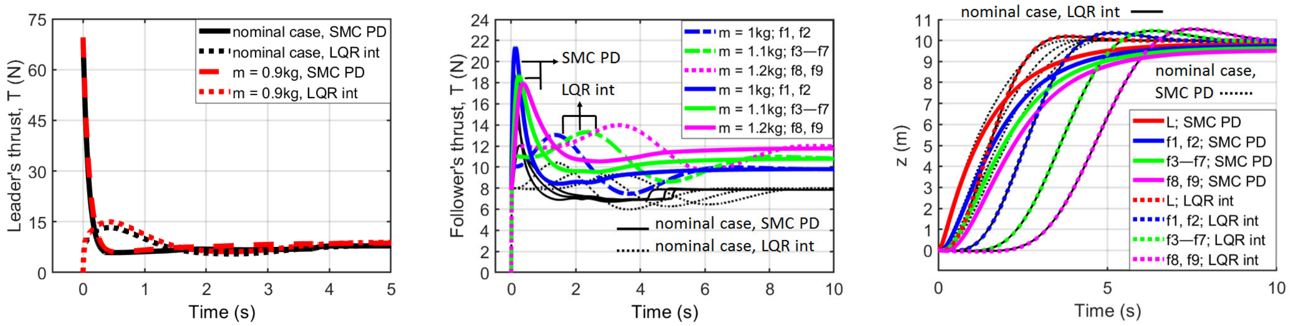
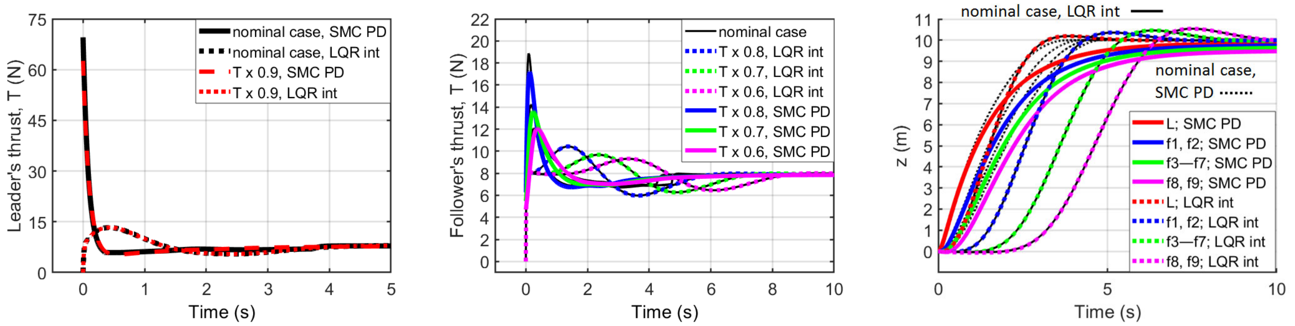


FIGURE 10. Kinetic vs. potential energies of the swarm.



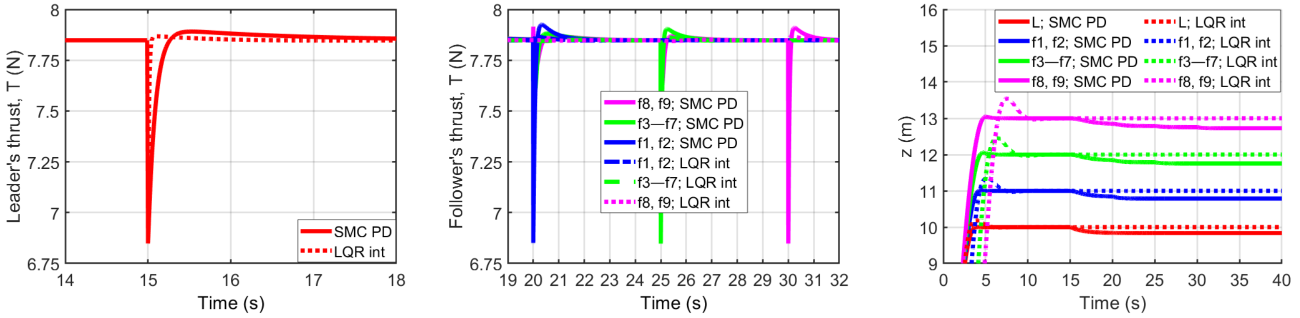
(a) Control action T of the leader (b) Control action T of the followers (c) Controlled output z

FIGURE 11. Results with the step change reference model, parametric uncertainties.



(a) Control action T of the leader (b) Control action T of the followers (c) Controlled output z

FIGURE 12. Results with the step change reference model, input uncertainties.



(a) Control action T of the leader (b) Control action T of the followers (c) Controlled output z

FIGURE 13. Results with the step change reference model, external disturbance inputs.

B. EXTERNAL DISTURBANCE INPUTS

In this section, a step change in thrust, $T_{real} = T + S$, $S = 0 \longrightarrow -1$N, is added as an external disturbance to the quadcopter(s). This is to be considered here that for the SMC PD control design, the external disturbance should be less than 0N for the stability of its altitude. On each level of the swarm, the external disturbance is added at time $t = \{15, 20, 25, 30\}$s respectively, and the simulation's running time t is 40s. Rest of the data for this test is given in Tables 1–3, and the obtained results are illustrated in Fig. 13. The results exhibit that the proposed nonlinear and linear, i.e. the SMC PD and the LQR integral action respectively, control designs provide the efficient stability of the closed-loop system and reject the disturbances quite well. For a clear depiction, the z positions, $\{z_L, z_{f1,f2}, z_{f3–f7}, z_{f8,f9}\}$, are plotted at the heights of $\{10, 11, 12, 13\}$m respectively in Fig. 13(c) although all setpoints are set to 10m. Similar to section VI-A, the performance of the LQR integral action control design is seen better due to the elimination of the tracking errors w.r.t. their settling points in the altitudes. However, fast response time is achieved using the SMC PD control strategy.

VII. CONCLUDING REMARKS

In this paper, the model of a swarm of drones that are arranged in the leader–follower architecture having hierarchical levels for a tightly coupled formation flight is proposed. This architecture is modular in the sense that it divides the overall system into four layers forming a hierarchical swarm: top-level leader drone, mid-level drones, and end-level followers.

For the evaluation of the control architecture, the distributed nonlinear and linear control methodologies comprising of SMC PD and LQR integral action designs respectively are applied on each drone. The control systems are tested for the tracking of a step change, an 8 curve, and an arithmetic spiral. The side by side comparative analysis based on setpoint tracking, response and settling time, collision avoidance, and computational costs is presented. It is evident from the results that both the proposed control designs are effective w.r.t. their nominal case, plant-model mismatch, and external disturbance inputs. In the nominal case, the SMC PD control design avoids a large deviation of the individual drone from its optimal navigation track due to its optimized transient behaviour in comparison to the LQR integral action technique. Also, the nominal settling time of the swarm is improved by 44% on average. Since a slight change in the system's dynamics can affect the overall system's state due to hierarchical and tightly coupled nature of the system, both the proposed approaches are tested for the cases of plant-model mismatch and external disturbance inputs. It is confirmed from the results of these cases that the performance using the LQR integral action control design is better than that using the SMC PD control strategy w.r.t. their settling time. One reason for the LQR to perform better in this case is that it includes integral action that is good for the elimination of unknown disturbances and model errors. Contrarily, the SMC uses the model in the calculation of the feedback law and is thus dependent on its PD structure while the LQR is a constant feedback of the real states. Satisfactory performances are achieved using both the proposed distributed control methodologies for the drones that are arranged in the leader–follower architecture having hierarchical levels for a tightly coupled formation flight. Both the approaches are scalable and effective as well as develop the versatility for an emerging area of UAVs.

APPENDIX

A. SMC PD CONTROL SYSTEM

$$\begin{aligned}
K_\alpha &\geq \{\frac{m}{cos\theta cos\phi}, J_x, \frac{J_y}{cos\phi}, \frac{J_z cos\theta}{cos\phi}\}\\
\beta_\alpha &= \{0.1, 0.1, 0.1, 0.1\}\\
c_\alpha &= \{0.25, 4, 8, 0.5\}
\end{aligned}$$

TABLE 9. Control gains for the PD blocks.

Position	Gain	L	f1, f2	f3–f7	f8, f9
x	P	-0.155	-0.11	-0.1	-0.3
	D	-0.2	-0.376	-0.12	-0.25
	Compensator	none	none	$\frac{s+0.01}{s+1}$	$\frac{s+0.01}{s+1}$
y	P	0.06	0.14	0.2	0.2
	D	0.127	0.176	0.25	0.3
	Compensator	none	$\frac{s+0.01}{s+1}$	$\frac{s+0.01}{s+1}$	$\frac{s+0.01}{s+1}$
z	P	3.085	10.3	15	17
	D	0.66	2.1	5	5
	Compensator	none	none	none	none

B. LQR INTEGRAL ACTION CONTROL SYSTEM

$$\begin{aligned}
G_r &= \{G_{x_r}, G_{y_r}, G_{z_r}\} = \{1, 1, \frac{1}{0.57s+1}\}\\
Q &= \mathrm{diag}\left(\begin{bmatrix} 3, 4.5, 6000, 1080, 1080, 1080, 180, 180,\\ 180, 0.5, 0.75, 1000, 15, 22.5, 30000 \end{bmatrix}\right)\\
R &= I_4
\end{aligned}$$

ACKNOWLEDGMENT

The authors would like to thank FinELib consortium, Finland for the APC funding, and Prof. H. Toivonen, Laboratory of Process and Systems Engineering, Åbo Akademi University, Turku, Finland, for discussions related to this article.

REFERENCES

[1] A. Tahir, J. Böling, M.-H. Haghbayan, H. T. Toivonen, and J. Plosila, "Swarms of unmanned aerial vehicles—A survey," *J. Ind. Inf. Integr.*, vol. 16, Oct. 2019, Art. no. 100106.

[2] P. Wang and F. Hadaegh, "Coordination and control of multiple microspacecraft moving in formation," *J. Astron. Sci.*, vol. 44, no. 3, p. 315–355, 1996.

[3] D. Galzi and Y. Shtessel, "UAV formations control using high order sliding modes," in *Proc. Amer. Control Conf.*, 2006, p. 4249.

[4] B. Yun, B. M. Chen, K. Y. Lum, and T. H. Lee, "Design and implementation of a leader-follower cooperative control system for unmanned helicopters," *J. Control Theory Appl.*, vol. 8, no. 1, pp. 61–68, Feb. 2010.

[5] M. Lewis and K. Tan, "High precision formation control of mobile robots using virtual structures," *Auto. Robots*, vol. 4, no. 4, p. 387–403, Oct. 1997.

[6] T. Paul, T. R. Krogstad, and J. T. Gravdahl, "Modelling of UAV formation flight using 3D potential field," *Simul. Model. Pract. Theory*, vol. 16, no. 9, pp. 1453–1462, Oct. 2008.

[7] Z. Chao, S.-L. Zhou, L. Ming, and W.-G. Zhang, "UAV formation flight based on nonlinear model predictive control," *Math. Problems Eng.*, vol. 2012, pp. 1–15, Feb. 2012.

[8] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Trans. Robot. Autom.*, vol. 14, no. 6, pp. 926–939, Dec. 1998.

[9] J. R. T. Lawton, R. W. Beard, and B. J. Young, "A decentralized approach to formation maneuvers," *IEEE Trans. Robot. Autom.*, vol. 19, no. 6, pp. 933–941, Dec. 2003.

[10] D. Bennet and C. McInnes, "Verifiable control of a swarm of unmanned aerial vehicles," *Proc. Inst. Mech. Eng., G, J. Aerosp. Eng.*, vol. 223, no. 7, p. 939–953, 2009.

[11] S. Engebraten, K. Glette, and O. Yakimenko, "Field-testing of high-level decentralized controllers for a multi-function drone swarm," in *Proc. IEEE 14th Int. Conf. Control Autom. (ICCA)*, Jun. 2018, p. 379.

[12] T. F. K. Cordeiro, H. C. Ferreira, and J. Y. Ishihara, "Non linear controller and path planner algorithm for an autonomous variable shape formation flight," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2017, p. 1493.

[13] B. Niemoczynski, S. Biswas, J. Kollmer, and F. Ferrese, "Hovering synchronization of a fleet of quadcopters," in *Proc. 7th Int. Symp. Resilient Control Syst. (ISRCS)*, Aug. 2014, p. 1–5.

[14] S. Srigrarom, H. X. Lin, Z. Y. Saw, J. Zhang, and C. H. Lim, "Design and build of swarm quadrotor UAVs at UGS," in *Proc. 15th AIAA Aviation Technol., Integr., Oper. Conf.*, Jun. 2015, p. 238.

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

[16] P. Lucas, K. Loayza, and E. Pelaez, "A distributed control of movements and fuzzy logic-based task allocation for a swarm of autonomous agents," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2018, p. 1–8

[17] H. M. H. Abdoli, M. Najafi, I. Izadi, and F. Sheikholeslam, "Sliding mode approach for formation control of multi-agent systems with unknown nonlinear interactions," *ISA Trans.*, vol. 80, pp. 65–72, Sep. 2018.

[18] K. Choutri, M. Lagha, L. Dala, and M. Lipatov, "Quadrotors UAVs swarming control under leader-followers formation," in *Proc. 22nd Int. Conf. Syst. Theory, Control Comput. (ICSTCC)*, Oct. 2018, p. 794.

[19] K. A. Ghamry and Y. Zhang, "Formation control of multiple quadrotors based on leader-follower method," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2015, p. 1037.

[20] F. Šolc, "Modelling and control of a quadrocopter," *Adv. Mil. Technol.*, vol. 5, no. 2, p. 29–38, 2010.

[21] P. Gabrlik, V. Kriz, and L. Zalud, "Reconnaissance micro uav system," *Acta Polytechnica CTU Proc.*, vol. 2, no. 2, pp. 15–21, 2015.

[22] A. Basci, K. Can, K. Orman, and A. Derdiyok, "Trajectory tracking control of a four rotor unmanned aerial vehicle based on continuous sliding mode controller," *Elektronika ir Elektrotechnika*, vol. 23, no. 3, pp. 12–19, 2017.

[23] Q. Quan, *Introduction to Multicopter Design Control*. Singapore: Springer, 2017.

[24] D. A. Mercado, R. Castro, and R. Lozano, "Quadrotors flight formation control using a leader-follower approach," in *Proc. Eur. Control Conf. (ECC)*, Jul. 2013, pp. 3858–3863.

[25] J. J. Castillo-Zamora, K. A. Camarillo-Gómez, G. I. Pérez-Soto, and J. RodriGuez-ReséNdiz, "Comparison of PD, PID and sliding-mode position controllers for V–Tail quadcopter stability," *IEEE Access*, vol. 6, pp. 38086–38096, 2018.

[26] J.-J. E. Slotine and W. Li, *Appl. Nonlinear Control*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1991.

[27] M. Safonov and M. Athans, "Gain and phase margin for multiloop LQG regulators," *IEEE Trans. Autom. Control*, vol. 22, no. 2, pp. 173–179, Apr. 1977.

[28] Q. Ali and S. Montenegro, "Explicit model following distributed control scheme for formation flying of mini UAVs," *IEEE Access*, vol. 4, pp. 397–406, 2016.

[29] C. Masse, O. Gougeon, D.-T. Nguyen, and D. Saussie, "Modeling and control of a quadcopter flying in a wind field: A comparison between LQR and structured $H\infty$control techniques," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2018, pp. 1408–1417.

ANAM TAHIR received the B.S. degree in computer engineering, the M.S. degree in electrical engineering (major: control systems) from COMSATS University, Islamabad, Pakistan, and the Master of Engineering degree in autonomous maritime operations from the NOVIA University of Applied Sciences, Turku, Finland. She is currently pursuing the Ph.D. degree with the Department of Future Technologies, Faculty of Science and Engineering, University of Turku, Finland. Her research interests include autonomous systems, adaptive and nonlinear control, modeling, and simulation of dynamical systems.

JARI M. BÖLING received the M.Sc. degree in chemical engineering and the Ph.D. degree in control engineering from Åbo Akademi University, Turku, Finland, in 1994 and 2001, respectively. From 2003 to 2004, he was a Postdoctoral Researcher with the University of California Santa Barbara, USA. Since 2005, he has been a Senior Lecturer of control engineering with the Faculty of Science and Engineering, Åbo Akademi University. His research interests include system identification, adaptive control, machine learning, modeling and simulation of dynamical systems.

MOHAMMAD-HASHEM HAGHBAYAN (Member, IEEE) received the B.A. degree in computer engineering from the Ferdowsi University of Mashhad, the M.S. degree in computer architecture from the University of Tehran, Iran, and the Ph.D. degree (Hons.) from the University of Turku, Finland. Since 2018, he has been a Postdoctoral Researcher with the Department of Future Technologies, Faculty of Science and Engineering, University of Turku. His research interests include machine learning, autonomous systems, high-performance energy-efficient architectures, and on-chip/fog resource management.

JUHA PLOSILA (Member, IEEE) is currently a Full Professor in autonomous systems and robotics with the Department of Future Technologies, Faculty of Science and Engineering, University of Turku (UTU), Finland, and the Ph.D. degree in electronics and communication technology from UTU, in 1999. He is the Head of the EIT Digital Master Program in embedded systems with the EIT Digital Master School, European Institute of Innovation and Technology, and represents UTU in the Node Strategy Committee of the EIT Digital Helsinki/Finland node. He has a strong research background in adaptive multiprocessing systems and platforms, and their design, including, specification, development and verification of self-aware multiagent monitoring and control architectures for massively parallel systems, machine learning and evolutionary computing-based approaches, as well as application of heterogeneous energy efficient architectures to new computational challenges in the cyber-physical systems and internet-of-things domains, with a recent focus on fog/edge computing (edge intelligence), and autonomous multidrone systems.