

Article

Genetic Optimization-Based Consensus Control of Multi-Agent 6-DoF UAV System

Aws Abdulsalam Najm ¹, Ibraheem Kasim Ibraheem ¹, Ahmad Taher Azar ^{2,3,*}
and Amjad J. Humaidi ⁴

¹ Department of electrical engineering, College of engineering, University of Baghdad, Baghdad 10001, Iraq; aws.najm@coeng.uobaghdad.edu.iq (A.A.N.); ibraheemki@coeng.uobaghdad.edu.iq (I.K.I.)

² Robotics and Internet-of-Things Lab (RIOTU), Prince Sultan University, Riyadh 11586, Saudi Arabia

³ Faculty of Computers and Artificial Intelligence, Benha University, Benha 13518, Egypt

⁴ Department of Control and Systems Engineering, University of Technology, Baghdad 10001, Iraq; 601116@uotechnology.edu.iq

* Correspondence: aazar@psu.edu.sa

Received: 20 April 2020; Accepted: 17 June 2020; Published: 24 June 2020



Abstract: A consensus control law is proposed for a multi-agent system of quadrotors with leader–follower communication topology for three quadrotor agents. The genetic algorithm (GA) is the proposed optimization technique to tune the consensus control parameters. The complete nonlinear model is used without any further simplifications in the simulations, while simplification in the model is used to theoretically design the controller. Different case studies and tests are done (i.e., trajectory tracking formation and switching topology) to show the effectiveness of the proposed controller. The results show good performance in all tests while achieving the consensus of the desired formations.

Keywords: hybrid control system; nonlinear PID; active disturbance rejection control; quadrotor system; unmanned aerial vehicle

1. Introduction

In the last decade, unmanned aircrafts (UAs) have attracted researchers in different technical and scientific communities. UAs have a wide range of applications in various fields, such as agriculture, healthcare, and entertainment. Due to the great variety of UA types, UAs then can be classified by their application, range, altitude, endurance, vehicle type, and size. Many UA controlling missions are difficult, and some are impossible to accomplish using only a single agent UA, but instead a cooperative control must be designed for multiple agents to accomplish these missions. Interest in cooperative control of multi-agent systems (MASs) has recently increased, because of its broad applications, such as fire detection [1]. A literature survey [2] summarized some of the potential applications of the multi-agent system, such as large object transportation, which was done in the GRASP laboratory at the University of Pennsylvania, surveillance and searching of objects, quadrotor localization in outdoor environment, which was done at the Czech Technical University, and object transportation missions [3]. There are many cooperation control problems, such as consensus control, formation control, flocking control, and so on. Another important research field is optimization, which has attracted many researchers due to the advantages provided by optimization, especially with complex and nonlinear systems. There has been much research in this field with different optimization techniques, such as genetic algorithm (GA) optimization [4,5], particle swarm optimization (PSO) [6], ant colony optimization (ACO) [7], as well as hybridization of two or more techniques [8,9].

1.1. Multi-Agent Systems

Generally, MASs are systems composed of multiple cooperating agents. Two significant abilities exist in agents that are controlled by computers (computerized agents). Firstly, they have the ability to do an independent task (i.e., deciding what to do to fulfill their task goals). Secondly, they have the ability to communicate with each other (i.e., they not only swap data, but they also participate with their peers by negotiation, cooperation, coordination, and so on) [10]. As mentioned earlier, there are many cooperative control problems. In this paper, light will be shed on the most common cooperating control problems, i.e., consensus control and formation control.

1.1.1. Consensus Control

Consensus problem is closely related to formation control (forming a specific geometrical shape, which is discussed next), flocking (a behavior presented when a group of birds are foraging or in-flight), and distributed estimation [11]; thus, it is an important and essential problem in the field of cooperative control. Consensus means reaching an agreement between multiple agents to achieve a common final state by communicating with each other through a communication network [12]. Figure 1 shows the consensus problem for four agents in their elevations only, where the four agents on the left side have different elevations above the ground (i.e., h_1 , h_2 , h_3 , and h_4) before applying the consensus control algorithm, while the agents on the right hand side present the agents after applying the consensus controller and reaching consensus in the elevation (i.e., h).

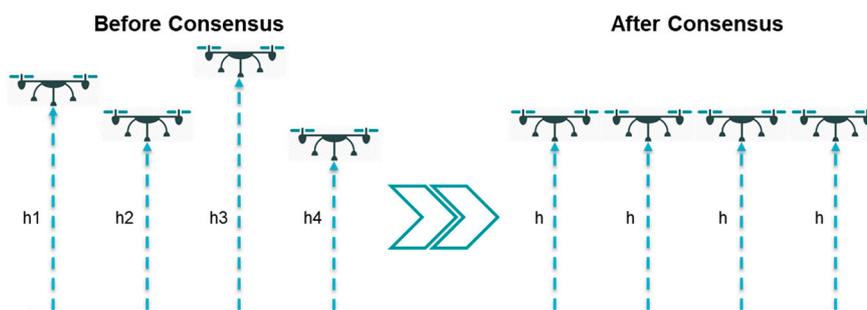


Figure 1. Consensus problem.

1.1.2. Formation Control

The formation control problem requires that all the agents reach a specific geometrical formation; thus, formation control differs from consensus by adding a distance to achieve the geometrical shape by all the agents; therefore, the formation control can be varied into consensus control by making the additional distance equal to zero. There are two cases for the formation control problem according to information known to agents. The first case is where the desired formation and its location are known for the team agents (i.e., centralized case). The second case is where the desired formation is known for the agents, while its location must be achieved by the team agents after negotiation [13]. Figure 2 shows five agents before and after applying formation control. In the initial positions, the agents are randomly distributed in two dimensions, and the heights of the agents are not taken into consideration. A formation controller is applied to achieve a triangle geometry shape, which is presented in the final formation.

Consensus and formation control algorithms can be categorized into leaderless and leader–follower depending on how the agents are connected in the communication network represented using graph theory [14].

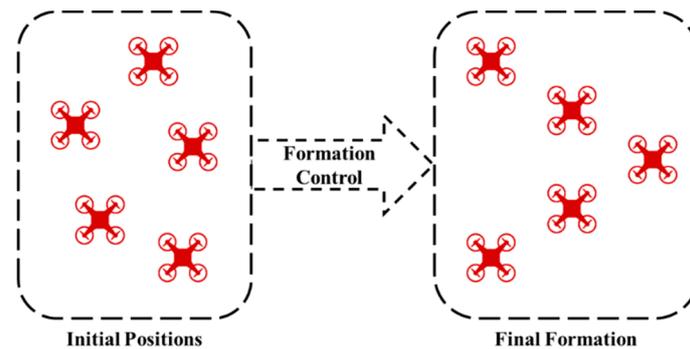


Figure 2. Before and after applying formation control on 5 agents.

1.1.3. Optimization Techniques

Finding the optimum performance index by tuning the controller parameters without any constraint is the purpose of the optimization techniques in this paper; therefore, the optimization technique used must be unconstrained. There are several methods to solve an unconstrained minimization problem, which can be classified as nongradient methods (e.g., random search method, grid search method, simplex method, etc.) and gradient methods (e.g., Fletcher–Reeves method, Newton’s method, Marquardt method, etc.).

For simple problems that involve a small number of variables, the nongradient methods are efficient, while the gradient methods require the first and sometimes the second derivative of the objective function (performance index). Both methods are not efficient in the case of this work due to the number of variables available in the tuning process and the complexity gained by getting the first or second derivative of the performance index used. Therefore, the modern optimization techniques are the best choice for the tuning and optimization problem of the proposed controller.

There is a wide range of modern methods available, such as genetic algorithms, simulated annealing, PSO, ACO, etc., which can be used in the tuning process. Due to the ability of the GA to avoid local optimal solutions, it was chosen as the optimization method to tune the parameters of the proposed controller.

1.1.4. Genetic Algorithm

The genetic algorithm (GA) is categorized as an evolutionary algorithm that is suitable in searching for a global solution. Inspiration in GA development comes from natural genetics working principles. Three main operators are used at each step to produce the next generation from the current one. These three operators are selection, crossover, and mutation, which are briefly explained in the following [4]:

- (1) Selection: The selection operation forms a mating pool for the next generation by selecting genes (parents) from the existing generation.
- (2) Crossover: The crossover process combines two of the parents to form the next generation (children).
- (3) Mutation: The mutation forms new children by applying random changes to the parents.

A concise diagram for the GA steps is shown in Figure 3. In this paper, the MATLAB function `ga` was used in the optimization and tuning processes.

1.2. Literature Survey

There is a large variation of research on cooperative control techniques for MASs. Getting an agreement between the agents (i.e., consensus) was the aim of some of the researchers, while others intended to obtain a specific formation for the MAS under some constraints (i.e., communication topology, leader/leaderless, . . . , etc.). This depends on the application where the control technique is

used. Some of the recent research, such as authors in [15], proposed a distributed adaptive controller for a class of an under-actuated Lagrangian system to track a dynamic leader and control the actuated variables, while being subjected to parameter uncertainties and external disturbance with or without actuator faults. In [16], the authors studied the distributed consensus problem for a team of quadrotors with fixed and switching topologies after transforming the quadrotor model to a simplified one using feedback linearization technique. Furthermore, others in [17] presented a two-layer hierarchical model predictive control (MPC) to stabilize a quadrotors team subjected to some constraints. Authors in [18] designed an adaptive consensus controller based on multiple surface control (MSC) to achieve a specific leader–follower formation for a team of quadrotors, taking into consideration the uncertainties generated by the ground effects. The results presented showed good performance with good adaptation of the slack variables and the disturbance of landing or taking off. Five different controllers were developed in [19]. Two of them (i.e., the nonlinear suboptimal H_∞ control technique and the integral backstepping (IBS) controller) were based on Lyapunov theory and were proposed for trajectory tracking and leader–follower formation control. The others were proportional derivative square (PD^2), which was compared with H_∞ control by applying them on the attitude system and the linear quadratic regulator (LQR) with the optimal iterative LQR, which were applied and compared for attitude, trajectory tracking, and formation control. Researchers in [20] used a simplified quadrotor model to develop a novel distributed consensus algorithm, while one of the uniform constant delays and asynchronous time-varying delays existed as a communication delay. A backstepping based on proportional integral derivative (PID) control was designed in [21] to solve the formation problem for a team of quadrotors in a decentralized manner. The system stability and retention of the desired geometric formation with or without the existence of the obstacles were presented in the simulations. Authors in [22] proposed a novel consensus leader–follower formation control based on a non-smooth backstepping design. The different numerical simulations verified the effectiveness of the proposed controller, and theoretical analysis showed that all the quadrotors converged to the desired formation.

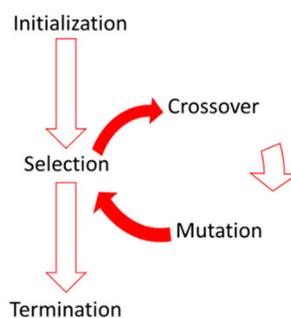


Figure 3. Steps diagram of the genetic algorithm (GA).

1.3. Motivation and Contributions

In the literature survey mentioned previously, a simplified model for the UA system was used in most of the research. This issue and others motivated us to simulate the complete UA quadrotor system model by using a proposed consensus controller based on optimization techniques. Moreover, the leader–follower topology was used in some case studies, where the leader agent was designed as a complete quadrotor system controlled by the Nonlinear PID (NLPID) and Improved Active Disturbance Rejection Control (IADRC) scheme proposed in our previous work, with the followers controlled by the controller scheme proposed in this work.

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

- Proposing a new consensus controlling scheme based on optimization techniques for a MAS of 6-degree of freedom (DOF) quadrotors.

- Simulating a leader–follower topology in some cases by adding additional terms to control the geometry of the agents, and if these terms are equalized to zero, the agents goes to the consensus case.

1.4. Notations

In this paper, vectors and matrices are denoted by bold symbols (e.g., \mathbf{A} or \mathbf{a}). Dot notation is used for time derivative. In this notation a dot is placed above the function name to denote the time derivative of a function (e.g., the first and second derivatives of x with respect to time t are denoted by \dot{x} , \ddot{x}). The integration in this paper is denoted by the symbol \int (e.g., the integration of the function x with respect to time t between a lower limit a and upper limit b is denoted by $\int_a^b x dt$). The sign \otimes is used in this paper as a Kronecker product operation.

1.5. Paper Organization

The rest of the paper is organized as follows: Some preliminaries on the quadrotor mathematical model, graph theory, mathematical representation of consensus control, and finally the problem statement of this paper are given in Section 2; thereafter, the proposed consensus control scheme is demonstrated in Section 3. Next, some simulations and tests are performed and discussed in Section 4. Finally, Section 5 includes the paper conclusions and some of the future aspects.

2. Preliminaries and Problem Statement

2.1. Quadrotor Mathematical Model

As mentioned in our previous works, the mathematical model of the quadrotor system is derived in Appendix A based on Newton–Euler equation and can be represented in Equation (1), while Newton’s law is used to obtain Equation (2), which is helpful to design a controller for the under-actuated quadrotor system using the acceleration in 3 dimensions [23,24]; all the parameters used are described in Table 1.

$$\left\{ \begin{array}{l} \dot{x} = c(\psi)c(\theta)u + [c(\psi)s(\phi)s(\theta) - c(\phi)s(\psi)]v + [s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta)]w \\ \dot{u} = rv - qw - gs(\theta) + \frac{f_{wx}}{m} \\ \dot{y} = c(\theta)s(\psi)u + [c(\phi)c(\psi) + s(\phi)s(\psi)s(\theta)]v + [c(\phi)s(\psi)s(\theta) - c(\psi)s(\phi)]w \\ \dot{v} = -ru + pw + gs(\phi)c(\theta) + \frac{f_{wy}}{m} \\ \dot{z} = -s(\theta)u + c(\theta)s(\phi)v + c(\phi)c(\theta)w \\ \dot{w} = qu - pv + gc(\theta)c(\phi) + \frac{f_{wz}-f_t}{m} \\ \dot{\phi} = p + s(\phi)t(\theta)q + c(\phi)t(\theta)r \\ \dot{p} = \frac{I_{yy}-I_{zz}}{I_{xx}}rq + \frac{\tau_x+\tau_{wx}}{I_{xx}} \\ \dot{\theta} = c(\phi)q - s(\phi)r \\ \dot{q} = \frac{I_{zz}-I_{xx}}{I_{yy}}pr + \frac{\tau_y+\tau_{wy}}{I_{yy}} \\ \dot{\psi} = \frac{s(\phi)}{c(\theta)}q + \frac{c(\phi)}{c(\theta)}r \\ \dot{r} = \frac{I_{xx}-I_{yy}}{I_{zz}}pq + \frac{\tau_z+\tau_{wz}}{I_{zz}} \end{array} \right. \quad \begin{array}{l} (a) \\ (b) \\ (c) \\ (d) \\ (e) \\ (f) \end{array} \quad (1)$$

$$\left\{ \begin{array}{l} \ddot{x} = -\frac{f_t}{m}[s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta)] \\ \ddot{y} = -\frac{f_t}{m}[c(\phi)s(\psi)s(\theta) - c(\psi)s(\phi)] \\ \ddot{z} = -\frac{f_t}{m}[c(\phi)c(\theta)] + g \end{array} \right. \quad (2)$$

Table 1. Parameter descriptions.

Parameters	Description	Units
$[x \ y \ z]$	Linear position vector	m
$[\phi \ \theta \ \psi]$	Angular position vector	Rad
$[u \ v \ w]$	Linear velocity vector	m/s
$[p \ q \ r]$	Angular velocity vector	Rad/s
$[I_{xx} \ I_{yy} \ I_{zz}]$	Moment of inertia vector	kg·m ²
f_i	Total thrust generated by rotors	N
$[\tau_x \ \tau_y \ \tau_z]$	Control torques	N m
$[f_{wx} \ f_{wy} \ f_{wz}]$	Wind force vector	N
$[\tau_{wx} \ \tau_{wy} \ \tau_{wz}]$	Wind torque vector	N m
g	Gravitational force	m/s ²
m	Total mass	kg
$c(\cdot) \equiv \cos(\cdot)$, $s(\cdot) \equiv \sin(\cdot)$, and $t(\cdot) \equiv \tan(\cdot)$		

2.2. Graph Theory

The mathematical preliminaries of the consensus control are important to be studied and understood to pave the way for dealing with the consensus control. The consensus control design procedure consists of a number of consecutive steps, which are listed below:

- Determining the number of agents.
- Deriving a graph based on agent communication.
- Deriving an adjacency matrix from the graph.
- Converting the adjacency matrix into a Laplacian matrix.
- Using the Laplacian matrix to design the consensus controller.

Thus, graph theory plays a key role in these steps, since it can reflect the communication topology of MASs [25].

Graph theory is commonly used in consensus control to describe the topology of the connection between the agents. Connection topology can be categorized as directed and undirected based on the information transmitted between the agents.

A graph \mathcal{G} is built upon a finite set, which has a limited number of elements. This finite set is called vertex set V , which includes the vertices of the graph. Assuming a graph \mathcal{G} has n vertices, then the vertex set V can be represented as $V = \{V_1, V_2, \dots, V_n\}$. A set of two vertices from V in the form $E = \{V_i, V_j\} \subset V \times V$, where $i, j = 1, 2, \dots, n$ and $i \neq j$, is called edge, while a set of edges is denoted by E . Sometimes another notation is used for the vertices and edges of the graph \mathcal{G} as $V(\mathcal{G})$ and $E(\mathcal{G})$, respectively, and for simplicity edge $\{V_i, V_j\}$ is denoted as $V_i V_j$.

The directed graph (digraph) is defined as a graph \mathcal{D} that consists of a vertex set $V(\mathcal{D})$ and directed edge set $E(\mathcal{D})$. In a directed graph, the direction of the edge is shown by an arrow. The graph is undefined if the vertex set is undefined.

2.2.1. Adjacency and Degree Matrices

For an undirected graph \mathcal{G} , each vertex V_i has a degree $d(V_i)$. This degree is equal to the number of vertices adjacent to V_i in \mathcal{G} . All the vertices' degrees of the graph are represented in matrix form as a diagonal matrix, which is called a degree matrix and is given as

$$\Delta(\mathcal{G}) = \begin{bmatrix} d(V_1) & 0 & \cdots & 0 \\ 0 & d(V_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d(V_n) \end{bmatrix} \quad (3)$$

where $\Delta(\cdot)$ is the degree matrix of graph \mathcal{G} , and n is the number of vertices or agents.

The adjacency matrix $A(\mathcal{G})$ is a symmetrical matrix that reflects adjacency links between the vertices in graph \mathcal{G} , and can be derived from the following relationship:

$$[A(\mathcal{G})]_{ij} = \begin{cases} 1 & \text{for } V_i V_j \in E(\mathcal{G}) \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

In all previously mentioned cases, the edge weights are considered to be one; but sometimes other weights are used instead of one. Generally, for weighted digraphs, the adjacency matrix is given as

$$[A(\mathcal{D})]_{ij} = \begin{cases} \varpi_{ij} & \text{for } (V_j, V_i) \in E(\mathcal{D}) \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where ϖ_{ij} is the weight for the edge between vertices V_i and V_j . The diagonal degree matrix is given as

$$\Delta(\mathcal{D}) = \begin{bmatrix} d_{in}(V_1) & 0 & \cdots & 0 \\ 0 & d_{in}(V_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_{in}(V_n) \end{bmatrix} \quad (6)$$

where $d_{in}(V_i)$ is the weighted in-degree of vertex V_i , which represents the sum of the weights of the edge that is coming from vertex V_j into vertex V_i and is given in the following relation:

$$d_{in}(V_i) = \sum_{\{j|(V_j, V_i) \in E(\mathcal{D})\}} \varpi_{ij} \quad (7)$$

2.2.2. Laplacian Matrix

There are many ways to represent the Laplacian of a graph. The most straight forward method is given as follows:

$$\mathcal{L}(\mathcal{G}) = \Delta(\mathcal{G}) - A(\mathcal{G}) \quad (8)$$

where $\Delta(\mathcal{G})$ is the degree matrix of graph \mathcal{G} , and $A(\mathcal{G})$ is the adjacency matrix for graph \mathcal{G} .

2.3. Mathematical Representation of Consensus Control

Consensus control stability and convergence analysis is a complex method due to the fact that consensus control is dealing with MAS, not a single agent. This is the reason why researchers in their publications simplify the system modeling to simplify the analysis approach, unlike the system modeling in a single agent. Some of the commonly used approaches for consensus control are the single-integrator system, double-integrator, and higher-order MASs, which are described next [25].

2.3.1. Single-Integrator MAS

MASs of single-integrator dynamics can be mathematically represented as follows:

$$\dot{x}_i = u_i, \quad i = 1, 2, \dots, N \quad (9)$$

where $x_i \in \mathbb{R}^n$ are the states vector for agent i , $u_i \in \mathbb{R}^m$ is the input vector for agent i , and N refers to the number of agents in the MAS. The consensus control is represented by

$$u_i = \mathcal{K} \sum_{j=1}^N \ell_{ij} (x_i - x_j) \quad (10)$$

where ℓ_{ij} is the (i, j) element of the Laplacian matrix, and $\mathcal{K} \in \mathbb{R}^{m \times n}$ is the control matrix, which needs to be determined.

To write the consensus in a general matrix form, substitute Equation (10) in Equation (9).

$$\dot{\mathbf{X}} = -\mathcal{K}\mathcal{L}(\mathcal{G})\mathbf{X} \quad (11)$$

where $\mathbf{X} = [x_1 \ x_2 \ \dots \ x_N]^T$, and $\mathcal{L}(\mathcal{G})$ is defined in Equation (8).

2.3.2. Double-Integrator MAS

MASs of double-integrator dynamics can be mathematically represented as

$$\begin{cases} \dot{x}_i = v_i \\ \dot{v}_i = u_i \end{cases}, \quad i = 1, 2, \dots, N \quad (12)$$

where $x_i, v_i \in \mathbb{R}^n$ are the position and velocity vectors, respectively, for agent i , and $u_i \in \mathbb{R}^m$ is the input vector for agent i . The consensus control is represented by

$$u_i = \mathcal{K} \sum_{j=1}^N \ell_{ij} ((x_i - x_j) + \gamma(v_i - v_j)) \quad (13)$$

The general matrix representation for the double-integrator MAS can be achieved by using Equation (12) with the controller in Equation (13) as follows:

$$\begin{bmatrix} \dot{\mathbf{X}} \\ \dot{\mathbf{V}} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{n \times n} & \mathbf{I}_n \\ -\mathcal{K}\mathcal{L}(\mathcal{G}) & -\gamma\mathcal{K}\mathcal{L}(\mathcal{G}) \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{V} \end{bmatrix} \quad (14)$$

where $\mathbf{X} = [x_1 \ x_2 \ \dots \ x_N]^T$ and $\mathbf{V} = [v_1 \ v_2 \ \dots \ v_N]^T$.

2.3.3. Higher-Order MAS

An extension for the single and double-integrator MASs is given as follows:

$$\begin{cases} \dot{x}_i = \mathbf{A}x_i + \mathbf{B}u_i \\ y_i = \mathbf{C}x_i \end{cases}, \quad i = 1, 2, \dots, N \quad (15)$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times m}$, $\mathbf{C} \in \mathbb{R}^{p \times n}$, $x_i \in \mathbb{R}^n$ is the state vector for agent i , $u_i \in \mathbb{R}^m$ is the input vector for agent i , and $y_i \in \mathbb{R}^p$ is the output vector for agent i . The consensus control can be achieved using the control defined in Equation (10) applied on the MAS of Equation (15). Then, the matrix form representation is as follows:

$$\dot{\mathbf{X}} = (\mathbf{I}_N \otimes \mathbf{A} - \mathcal{L}(\mathcal{G}) \otimes \mathbf{B}\mathcal{K})\mathbf{X} \quad (16)$$

2.4. Problem Statement

A network of homogenous connected MAS is considered as a leader–follower connection, where each agent mathematical model can be represented in the compact form. The control signal of the leader is treated as a single agent control problem, while the followers' controllers have to be designed and tuned based on the GA optimization technique such that they stabilize the unstable followers, reach consensus between all the MAS agents, and achieve desired formations while tracking different trajectories.

3. Proposed Controlling Scheme Design

The consensus control is proposed in this paper to achieve tracking formation for the MAS of the quadrotors while using the leader–follower connection topology between the agents (directed graph).

The aim of the tracking formation is to keep the agents in a formation on the 3D space (x, y, z) and the angle of rotation around z-axis (ψ) while tracking the leader. Therefore, the leader control is different from the followers, which can be represented by normal quadrotor control using a combination of NLPID for the outer loop control (OLC) and IADRC for the inner loop control (ILC) of the quadrotor system, which was designed in our previous works [26–28] and is shown in Figure 4. The followers’ control is a mix of two controllers; the first is the local one and the second is the consensus controller, as shown in Figure 5.

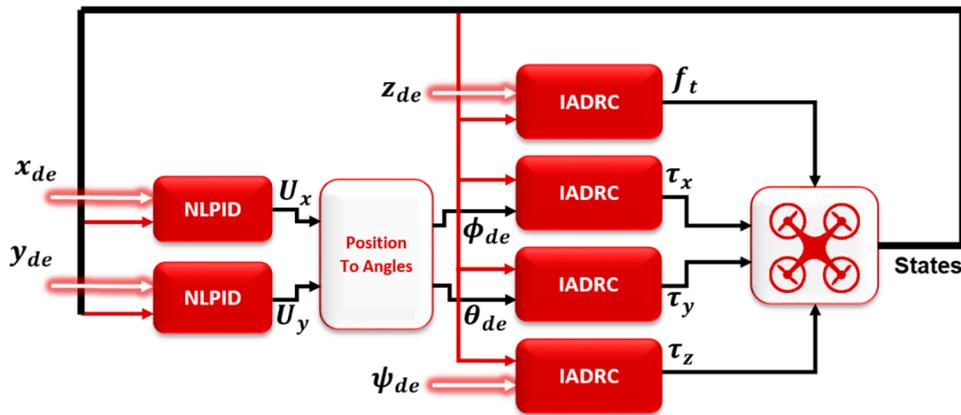


Figure 4. IADRC control for ILC and NLPID control for OLC of quadrotor system.

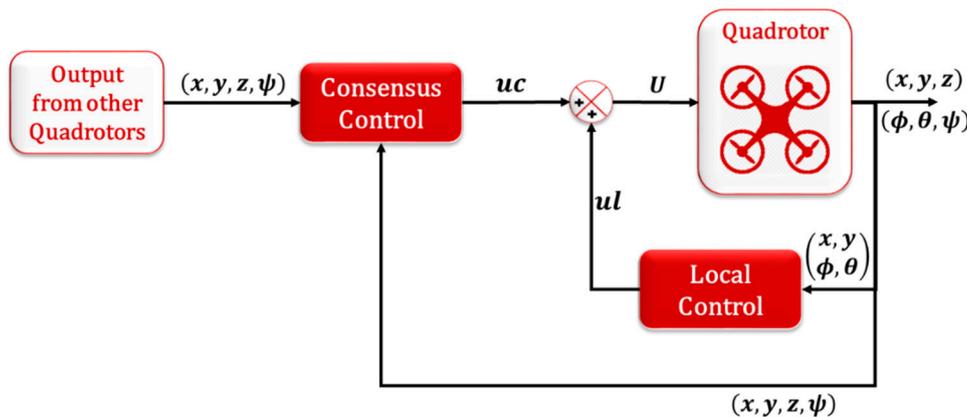


Figure 5. Local and consensus controllers.

The follower agent control system for (ϕ, θ) states are the IADRC designed for the leader agent, before proceeding in the proposed controller for the states (x, y, z, ψ) a brief explanation and equations for the NLPID and the IADRC are given.

3.1. Nonlinear PID (NLPID) Controller

The NLPID controller is developed based on the PID controller to be better while dealing with highly complicated nonlinear systems such as the 6-DOF quadrotor system. The NLPID controller is given in Equation (17).

$$\begin{cases} U_{NLPID} = f_1(e) + f_2(\dot{e}) + f_3(\int e dt) \\ f_i(\varepsilon) = k_i(\varepsilon)|\varepsilon|^{\alpha_i} \text{sign}(\varepsilon) \\ k_i(\varepsilon) = k_{i1} + \frac{k_{i2}}{1 + \exp(\mu_i \varepsilon^2)} \quad i \in \{1, 2, 3\} \end{cases} \quad (17)$$

where ε could be e, \dot{e} , or $\int e dt$, $\alpha_i \in \mathbb{R}^+$ and the function $k_i(\varepsilon)$ is a positive function with coefficients $k_{i1}, k_{i2}, \mu_i \in \mathbb{R}^+$. The term $k_i(\varepsilon)$ is introduced here to make the NLPID controller more sensitive for

small errors. Small errors which is approximate to zero, the value of the term $k_i(\varepsilon)$ approaches to the upper bound $k_{i1} + \frac{k_{i2}}{2}$, while for large errors, the term $k_i(\varepsilon)$ approaches the lower bound k_{i1} , this means that the limits of the term $k_i(\varepsilon)$ is bounded in the range $[k_{i1}, k_{i1} + \frac{k_{i2}}{2}]$.

3.2. Improved Active Disturbance Rejection Control (IADRC)

ADRC, is in general, a grouping of three dynamic systems: Extended State Observer (ESO), Tracking Differentiator (TD), and State Error Feedback (SEF) controller. In our previous paper, IADRC is proposed, which is a grouping of the abovementioned three dynamic systems taking into consideration that linear ESO (LESO) is used, TD is replaced by improved TD (ITD) and that SEF is changed by NLPID control.

3.2.1. Design of Linear Extended State Observer (LESO)

The key idea of this element is to convert the ESO with the error equation of the system to a disturbed asymptotical stable system, in which the high-gain eliminates the effect of the total disturbance error. The mathematical representation for the LESO used is given as,

$$\begin{cases} \dot{z}_1 = z_2 + \beta_1 e_1 \\ \dot{z}_2 = z_3 + \beta_2 e_1 \\ \vdots \\ \dot{z}_\rho = z_{\rho+1} + \beta_\rho e_1 + b_0 U \\ \dot{z}_{\rho+1} = \beta_{\rho+1} e_1 \end{cases} \quad (18)$$

$$U = u_0 - \frac{z_{\rho+1}}{b_0} \quad (19)$$

where u_0 is control input, $e_1 = (y - z_1)$, and β_i are gain constants of the observer needed to be tuned, $i \in \{1, 2, \dots, \rho, \rho + 1\}$. $\beta_i = c_i \omega_0$, $c_i, i \in \{1, 2, \dots, \rho, \rho + 1\}$ are relevant constants, and ω_0 is the bandwidth of the observer and need to be chosen to get better estimation for the state and the disturbance with minimum OPI, and $z_1, z_2, \dots, z_{\rho+1}$ represent the states of the plant estimated by the ESO.

3.2.2. Design of Improved Tracking Differentiator (ITD)

The aim of this element is to generate a smooth approximation of the reference signal and its derivatives for the control system, which can be mathematically represented as:

$$\begin{cases} \dot{r}_1 = r_2, r_1(0) = r_{10} \\ \dot{r}_2 = r_3, r_2(0) = r_{20} \\ \vdots \\ \dot{r}_\rho = -R^\rho \tanh\left(\frac{\beta r_1 - (1-\alpha)r}{\gamma}\right) - R^{\rho-1} r_2(t) - \dots - R r_\rho(t), \\ r_\rho(0) = r_{\rho 0} \end{cases} \quad (20)$$

where $\{r_1, r_2, \dots, r_\rho\}$ are the tracking signals (i.e., the reference signal and the derivatives of r) tracked by the differentiator signals and having initial values $\{r_{10}, r_{20}, \dots, r_{\rho 0}\}$. The parameters R, β, α and γ are the ITD parameters which have to be tuned to get the best tracking with small errors.

3.2.3. Design of Nonlinear PD Controller (NLPD)

The NLPID will be used as the SEF in the IADRC, which is the same in Equation (17), but for general chain of integrators system of order ρ . The general chain of integrators system can be mathematically represented as:

$$\begin{cases} u_0 = f_1(e) + f_2(\dot{e}) + \dots + f_\rho(e^\rho) + f_{\rho+1}\left(\int e dt\right) \\ f_i(\varepsilon) = k_i(\varepsilon)|\varepsilon|^{\alpha_i}\text{sign}(\varepsilon) \\ k_i(\varepsilon) = k_{i1} + \frac{k_{i2}}{1+\exp(\mu_i\varepsilon^2)}, i \in \{1, 2, \dots, \rho + 1\} \end{cases} \quad (21)$$

where ε could be $e, \dot{e}, \dots, e^\rho$ or $\int e dt$, and $(k_{i1}, k_{i2}, \mu_i, \alpha_i)$ are the controller parameters and are tuned to get minimum OPI. Due to the derivation in [24] of the IADRC, which changes the system into a chain of integrators, the integrator term in the NLPID will be neglected. The NLPD controller for general chain of integrators system of order ρ is represented as:

$$\begin{cases} u_0 = f_1(e) + f_2(\dot{e}) + \dots + f_\rho(e^\rho) \\ f_i(\varepsilon) = k_i(\varepsilon)|\varepsilon|^{\alpha_i}\text{sign}(\varepsilon) \\ k_i(\varepsilon) = k_{i1} + \frac{k_{i2}}{1+\exp(\mu_i\varepsilon^2)}, i \in \{1, 2, \dots, \rho\} \end{cases} \quad (22)$$

3.3. Consensus Control for $x - y$ Position Subsystems

The $x - y$ position dynamics of the follower agents are given by

$$\begin{cases} \ddot{x}_i = U_{xi} \\ \ddot{y}_i = U_{yi} \end{cases}, i = 1, 2, \dots, N \quad (23)$$

where N is the number of agents, while the leader agent is denoted by 0. The proposed controller for $x - y$ position subsystems can be represented as

$$\begin{cases} U_{xi} = ul_{xi} + uc_{xi} \\ U_{yi} = ul_{yi} + uc_{yi} \end{cases} \quad (24)$$

where ul and uc are the local and the consensus control signals for each agent, respectively. The consensus controller is designed as a single integrator system consensus controller, which is given by Equation (10) adding the formation position terms, while the local controller is designed to ensure that

$$\lim_{t \rightarrow \infty} \dot{x} = \lim_{t \rightarrow \infty} \dot{y} = 0$$

and is given by

$$\begin{cases} ul_{xi} = -k_x \dot{x}_i \\ ul_{yi} = -k_y \dot{y}_i \end{cases} \quad (25)$$

where k_x and k_y are the controllers' parameters, which need to be tuned and optimized to give better tracking performance. Then Equation (24) can be written as

$$\begin{cases} U_{xi} = -k_x \dot{x}_i + \mathcal{K}_x \sum_{j=1}^N \ell_{ij}((x_i - x_j - \rho_{xi})) \\ U_{yi} = -k_y \dot{y}_i + \mathcal{K}_y \sum_{j=1}^N \ell_{ij}((y_i - y_j - \rho_{yi})) \end{cases} \quad (26)$$

where ρ_{xi} and ρ_{yi} are the reference state in the x and y , respectively, and \mathcal{K}_x and $\mathcal{K}_y \in \mathbb{R}$ are the consensus control parameters, which need to be tuned and optimized to give better performance.

3.4. Consensus Control for z and ψ Subsystems

The dynamics of z subsystem is given as

$$\begin{cases} \dot{z} = -s(\theta)u + c(\theta)s(\phi)v + c(\phi)c(\theta)w \\ \ddot{z} = \dot{w} = -\frac{f_t}{m}[c(\phi)c(\theta)] + g \end{cases} \quad (27)$$

It is very clear that it is coupled and nonlinear. Therefore, to overcome this problem and design the consensus controller, the quadrotor system is assumed to be near the hovering point where $\phi = \theta \approx 0$. Then the dynamics of z subsystem is represented as follows:

$$\begin{cases} \dot{z} = w \\ \dot{w} = -\frac{f_t}{m} + g \end{cases} \quad (28)$$

which is a double integrator system and can be controlled using the consensus controller expressed in Equation (13), assuming that $\gamma = 1$ and adding the formation position terms. The controller of the z subsystem is given as

$$U_{zi} = f_t = \sum_{j=1}^N \ell_{ij} (\mathcal{K}_{z1}(z_i - z_j - \rho_{zi}) + \mathcal{K}_{z2}(w_i - w_j)) \quad (29)$$

while for the dynamics of ψ subsystem is given as

$$\begin{cases} \dot{\psi} = \frac{s(\phi)}{c(\theta)}q + \frac{c(\phi)}{c(\theta)}r \\ \dot{r} = \frac{I_{xx} - I_{yy}}{I_{zz}}pq + \frac{\tau_z + \tau_{wz}}{I_{zz}} \end{cases} \quad (30)$$

The assumption of the hovering point is still true and the inertias of the quadrotor are nearly equal (i.e., $I_{xx} \cong I_{yy} \cong I_{zz}$). Then, the dynamics of the ψ subsystem is represented as a double integrator in the following form:

$$\begin{cases} \dot{\psi} = r \\ \dot{r} = \frac{\tau_z}{I_{zz}} \end{cases} \quad (31)$$

which can be controlled using the consensus controller given in Equation (13); for simplicity, γ is assumed to equal one ($\gamma = 1$), because its effect will be reflected on the value of the consensus controller gain ($\mathcal{K}_{\psi 2}$) and adding the formation position terms. The controller of the ψ subsystem is given as

$$U_{\psi i} = \tau_z = \sum_{j=1}^N \ell_{ij} (\mathcal{K}_{\psi 1}(\psi_i - \psi_j - \rho_{\psi i}) + \mathcal{K}_{\psi 2}(r_i - r_j)) \quad (32)$$

where ρ_{zi} and $\rho_{\psi i}$ are the reference state in the z and ψ , respectively, and \mathcal{K}_{z1} , \mathcal{K}_{z2} , $\mathcal{K}_{\psi 1}$ and $\mathcal{K}_{\psi 2} \in \mathbb{R}$ are the consensus control parameters that need to be tuned and optimized to give better performance. The overall quadrotor system with the consensus controller is shown in Figure 6.

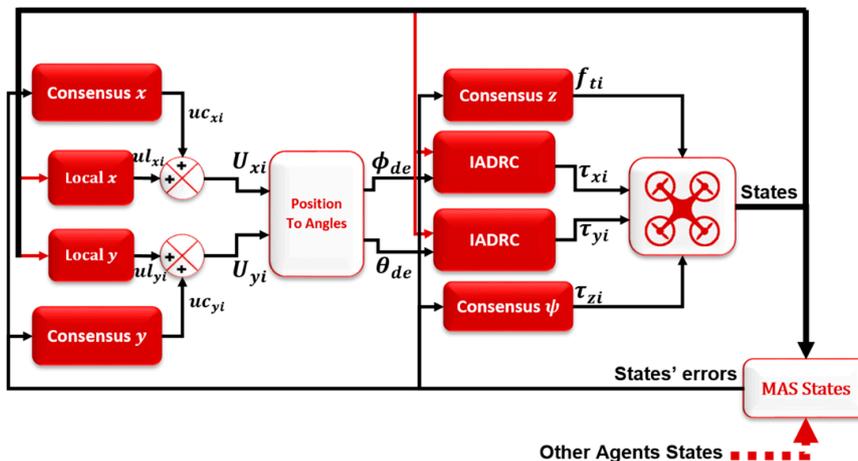


Figure 6. The overall ith quadrotor system with the consensus controller.

As shown in Figure 6, the local control units (i.e., IADRC) of the z and ψ subsystems were removed because the IADRC units keep tracking reference signals, which in this case must be zero signal. Therefore, the agents' states will be forced to reach zero. This causes canceling of the consensus control effect.

In contrast, the ϕ and θ subsystems do not have consensus controllers, while the local control units are still in place. This is due to the fact that the ϕ and θ subsystems have their own desired signals generated internally from the x and y position subsystems, as explained earlier, and can be treated as an internal part of the system. Therefore, adding consensus controllers will affect the x and y position subsystems and will never achieve the desired formation.

Remark 1. The linearized subsystems in Equation (23), Equation (28) and Equation (31) of the 6-DOF quadrotor are used only for theoretical design of the controller to choose the appropriate one, while the complete nonlinear model represented by Equation (1) is used in all the simulations.

4. Simulation Results and Discussion

In this section, a MAS of three quadrotors using the leader–follower topology for communication network was studied, as shown in Figure 7. All quadrotor agents' parameters values are defined in Table 2. The leader agent control system (NLPID–IADRC combination) parameters are given in Tables 3–6.

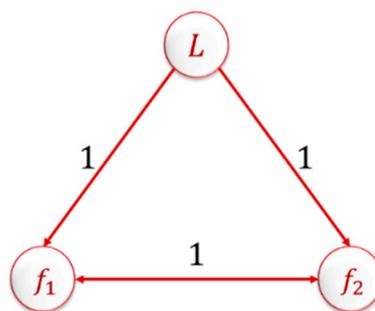


Figure 7. Three quadrotors leader–follower connection topology.

Table 2. Quadrotor model parameters.

Parameter	Value	Description	Unit
I_{xx}	8.5532×10^{-3}	Moment of inertia around x -axis	$\text{kg}\cdot\text{m}^2$
I_{yy}	8.5532×10^{-3}	Moment of inertia around y -axis	$\text{kg}\cdot\text{m}^2$
I_{zz}	1.476×10^{-2}	Moment of inertia around z -axis	$\text{kg}\cdot\text{m}^2$
g	9.81×10^0	Gravitational force	m/s^2
m	9.64×10^{-1}	Total mass	kg
b	7.66×10^{-5}	Thrust coefficient	$\text{N}\cdot\text{s}^2$
d	5.63×10^{-6}	Drag coefficient	$\text{N}\cdot\text{m}\cdot\text{s}^2$
l	2.2×10^{-1}	Motor to center length	m

Table 3. LESOs Parameters of the leader quadrotor.

	z	ϕ	θ	ψ
ω_0	300	861.36	671.76	749.05
b_0	0.5	0.0041	0.005	0.0047

Table 4. ITD parameters of the leader quadrotor.

z, ϕ, θ, ψ	
α	0.9789
β	2.7936
γ	16.7728
R	26.5005

Table 5. NLPD parameters of the leader quadrotor.

	z	ϕ	θ	ψ
k_{11}	32.4800	5.6390	0.51080	0.6993
k_{12}	11.4360	0.0764	0.0390	0.2109
k_{21}	9.0756	0.7495	0.0660	0.2415
k_{22}	0.1518	0.0471	0.0666	0.1024
μ_1	0.2814	0.0763	0.5191	0.1272
μ_2	0.4234	0.5997	0.7747	0.3760
α_1	0.9684	0.9591	0.9574	0.9741
α_2	0.9580	0.9547	1.0034	0.9417

Table 6. NLPID parameters for the $x - y$ position of the leader quadrotor.

	x	y		x	y
k_{11}	3.0735	2.3046	μ_1	0.1214	0.1057
k_{12}	0.5674	0.1108	μ_2	0.9859	0.9744
k_{21}	6.0204	4.5269	μ_3	0.7656	0.6008
k_{22}	2.0076	2.0040	α_1	0.9106	0.9101
k_{31}	2.2409×10^{-4}	2.3828×10^{-4}	α_2	0.9999	0.9994
k_{32}	5.5135×10^{-7}	6.8046×10^{-7}	α_3	0.9140	0.9137

The Laplacian matrix for the MAS based on the connection topology shown in Figure 7 is given as

$$\mathcal{L}(\mathcal{D}) = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

where the first row represents the leader agent, which does not receive any information from the followers; it just transmits its information to them. The weights used for the connection were ones. This represents that there was information transfer between the agents.

The single agent system has only one error for each state, while for the MAS the number of errors depends on the number of agents in the system. In the next simulations, for each state of the quadrotor, there were three errors (i.e., for state κ the errors were $\kappa_1 - \kappa_2$, $\kappa_1 - \kappa_3$, and $\kappa_2 - \kappa_3$ where $\kappa \in \{x, y, z, \psi\}$). For the reason mentioned previously, the multi-objective performance index (OPI) used here depended on the integral time absolute error (ITAE) performance index, which is given as

$$\begin{cases} opi_i = \sum_{j=1}^3 \omega_{ij} \times ITAE_{ij} \\ OPI = \sum_i (\hat{\omega}_i \times opi_i), i = x, y, z, \psi \end{cases} \quad (33)$$

where $ITAE_{ij}$ represents the $ITAE = \int_0^{t_f} |e(t)| dt$ for each of the three consensus errors. ω_{ij} was chosen to be $\frac{1}{3}$ for all i and j , while $\hat{\omega}_i$ was chosen to be 0.25 for all i .

The tuning process to obtain the optimum OPI was done using random initial values for $[x, y, z, \psi]$ states, such as $[0.001, 0.001, 0.001, 0]$, $[0.5, 0.5, 0.001, 0]$, and $[-0.5, -0.5, 0.001, 0]$ for the leader, the first

follower, and the second follower, respectively. The reference signals used were zero for x , y , and ψ , while a step input was used for the altitude (z), and taking into considerations ρ_{xi} , ρ_{yi} , ρ_{zi} , and $\rho_{\psi i}$ mentioned in Equation (26), Equation (29) and Equation (32), respectively, all were equal to zero during the tuning process.

All the values of the consensus control parameters are shown in Table 7. The performance indices and the OPI values after GA tuning are given in Table 8 based on the initial values and the reference signals mentioned.

Table 7. Consensus controller parameters.

x		y	
k_x	\mathcal{K}_x	k_y	\mathcal{K}_y
4.0259	0.9798	3.9732	1.0631
z		ψ	
\mathcal{K}_{z1}	\mathcal{K}_{z1}	$\mathcal{K}_{\psi 1}$	$\mathcal{K}_{\psi 2}$
14,169.9808	427.2904	13.8067	0.5709

Table 8. Performance indices and OPI values for consensus control.

	Consensus Control		
	ITAE1	ITAE2	ITAE3
x	1.076761	1.009015	1.879853
y	1.203705	1.177282	2.106002
z	0.838854	0.838702	0.001418
ψ	0.005238	0.005962	0.001104
OPI		0.845325	

Some tests were done on the MAS with the proposed consensus control scheme. These are represented as follows.

4.1. Achieving Consensus

The purpose of this test was to check the errors between the agents' states while following the leader agent and the time it took to reach consensus between the agents. The initial states for each agent are given in Table 9, while the reference signals applied for the altitude (z), position (x , y), and the Yaw angle (ψ) were zero signal with simulation time $t_f = 10$ s.

Table 9. Achieving consensus test initial states.

	Initial States			
	x	y	z	ψ
Leader	0.001	0.001	0.001	0
Follower 1	0.5	0.5	0.8	$\pi/4$
Follower 2	-0.5	-0.5	0.4	$-\pi/4$

Figures 8–11 a show the time response for the altitude (z), position (x , y), and the Yaw angle (ψ) states, while Figures 8–11 b show the errors between agents. The subscript L refers to the leader, while f refers to the followers. The smoothness in the time response for all cases was very clear with fast consensus-reaching (i.e., less than 4 s for the x and y position, while for the altitude z and the yaw angle ψ the time taken to reach consensus was less than 0.5 s). The errors occurring in all the states always converged to zero after a very short time even if the initial states were different.

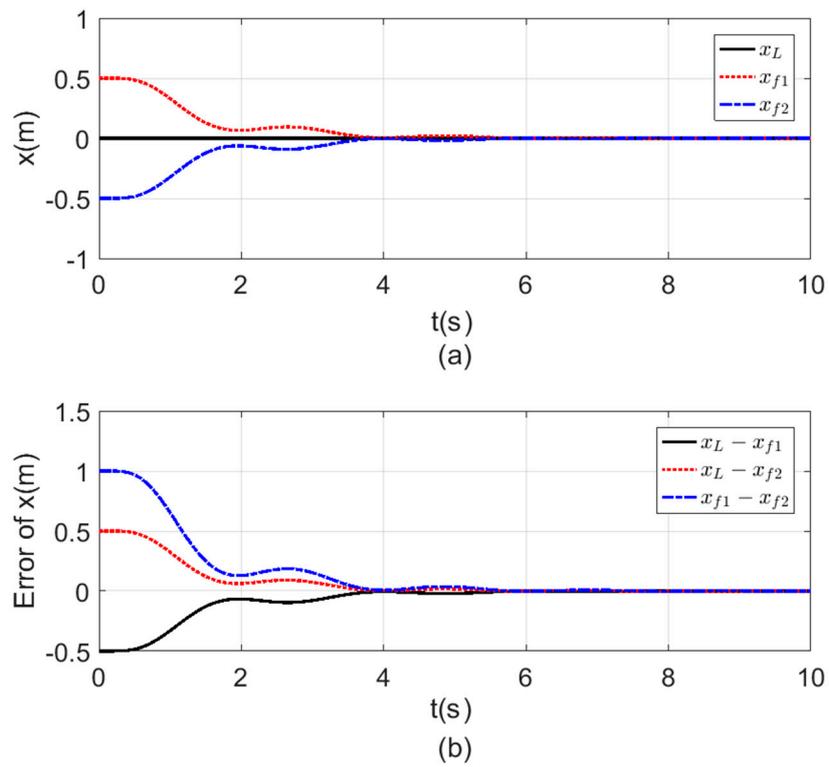


Figure 8. Consensus in x -position (a) x -position time response. (b) Error of the x -position.

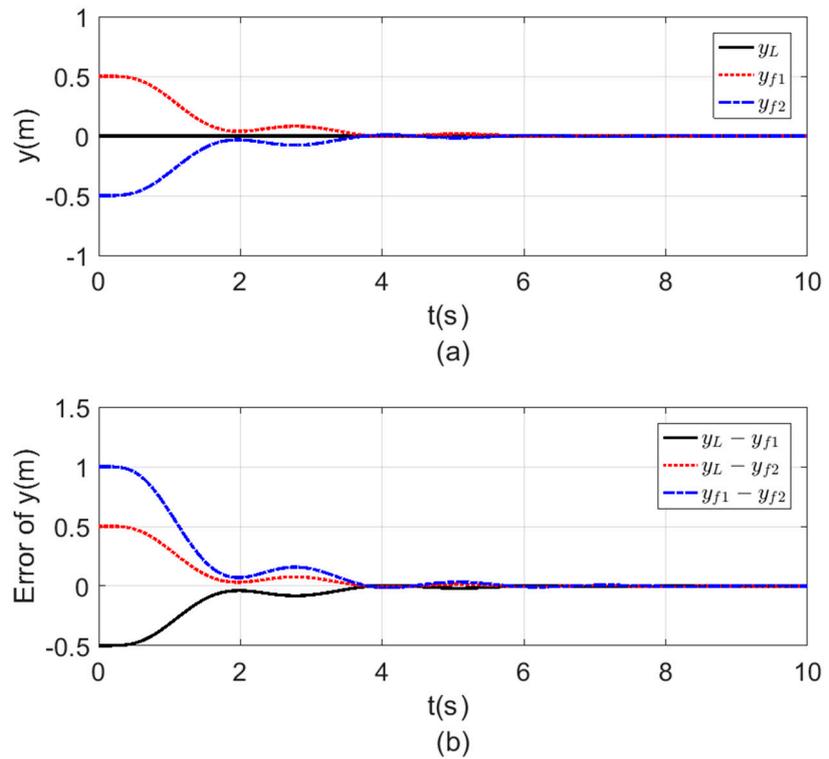


Figure 9. Consensus in y -position (a) y -position time response. (b) Error of the y -position.

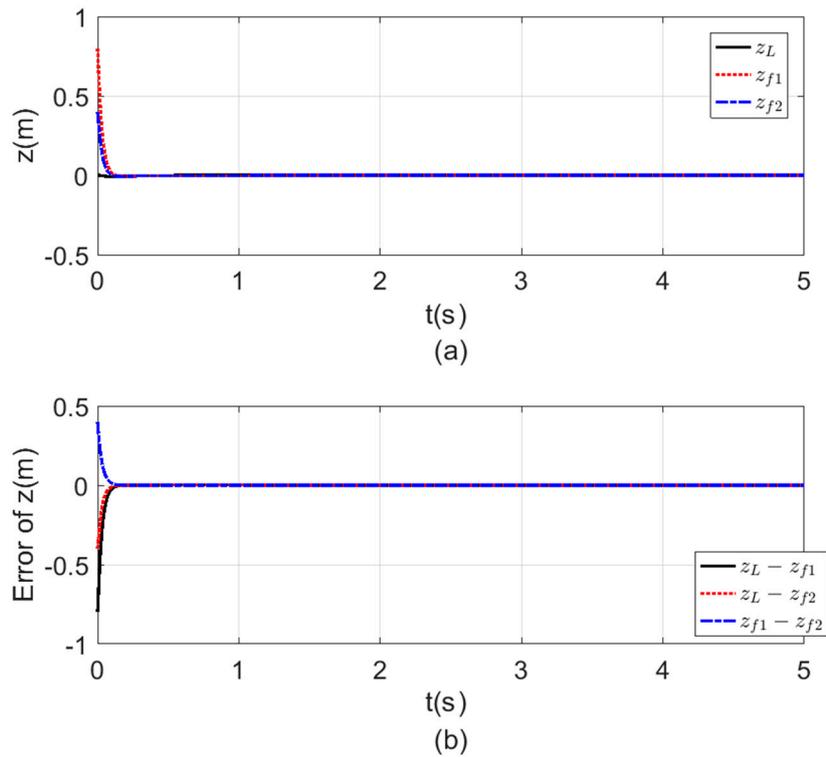


Figure 10. Consensus in z -position (a) z -position time response. (b) Error of the z -position.

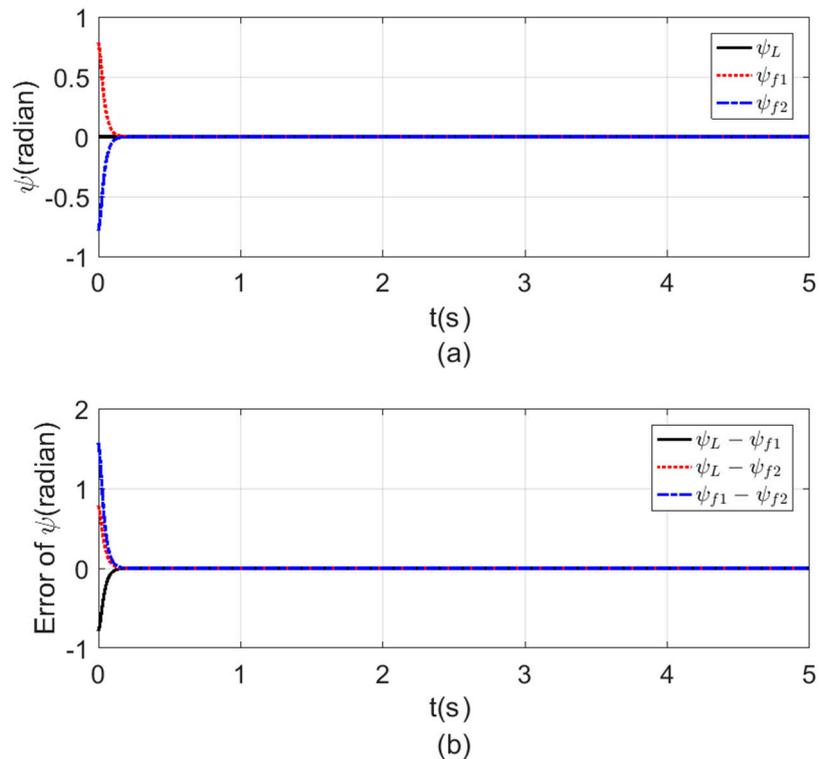


Figure 11. Consensus in ψ -angle (a) ψ -angle time response. (b) Error of the ψ angle.

4.2. Fixed Triangle Formation

A fixed formation for the agents was reached, while the leader was supposed to keep increasing in its altitude. The initial states for this test and the tests coming after were the same, as represented

in Table 9, taking into consideration that the followers' initial altitude was the same altitude as that of the leader (i.e., initial $z = 0.001$ for both followers). The reference signals of the leader for the x , y , and ψ were zero signal, while for z a ramp input was used with a slope = 0.2 m with simulation time $t_f = 30$. The values of the reference states to form the triangle shape (ρ_{xi} , ρ_{yi} , ρ_{zi} , and $\rho_{\psi i}$) mentioned in Equation (26), Equation (29) and Equation (32), respectively, are given as

$$\rho_{xL} = 0, \rho_{xf_1} = 1, \rho_{xf_2} = -1; \rho_{yL} = 0, \rho_{yf_1} = \rho_{yf_2} = 1;$$

$$\rho_{zL} = \rho_{zf_1} = \rho_{zf_2} = 0; \rho_{\psi L} = \rho_{\psi f_1} = \rho_{\psi f_2} = 0.$$

The time response for the $x - y$ position and the altitude z of the leader and the two followers are shown in Figure 12, while the motion of the agents in 3D is represented in Figure 13. From the results represented in Figure 12, the followers tracked the leader trajectory perfectly with fast response, keeping the distances given previously to form the triangle shape, while the altitude of the followers matched the leader path without a variation in the response. The motion in Figure 13 of the agents shows the agents, while changing their positions to form the triangle shape, which was formed by the three agents after a few seconds from starting the test, and the shape was kept fixed while the leader altitude was increasing.

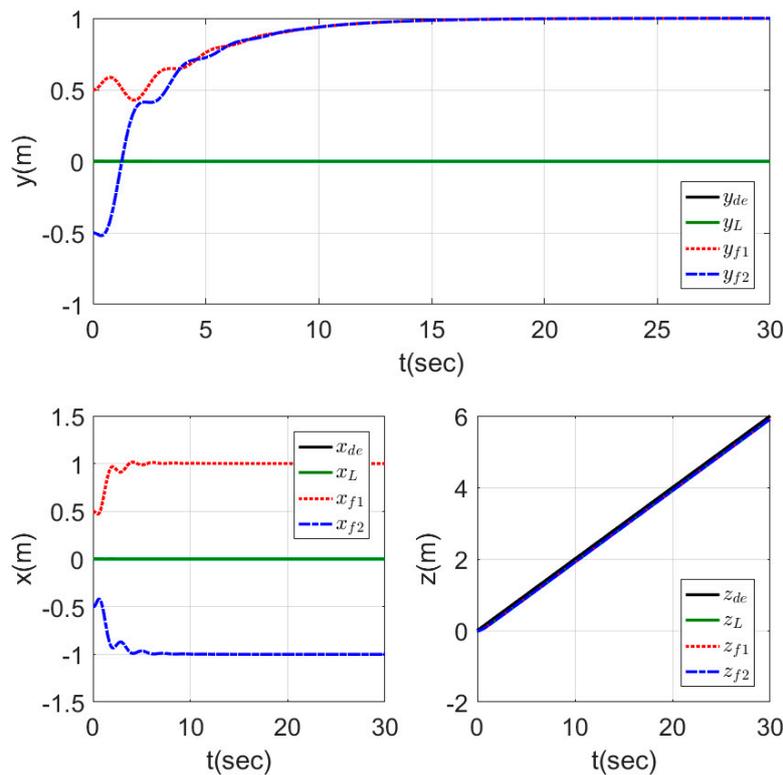


Figure 12. Time response of fixed triangle formation for MAS.

4.3. Fixed Formation Tracking Ramp Trajectory

This test differed from the previous one by the reference signals used with the leader to track them, where the (x and y) position was a changeable not zero signal. Forming a fixed triangle shape while the agents were moving was required in this test. The reference states values for the triangle shape are given as follows:

$$\rho_{xL} = \rho_{xf_1} = 0, \rho_{xf_2} = -1; \rho_{yL} = \rho_{yf_2} = 0, \rho_{yf_1} = -1;$$

$$\rho_{zL} = \rho_{zf_1} = \rho_{zf_2} = 0; \rho_{\psi L} = \rho_{\psi f_1} = \rho_{\psi f_2} = 0.$$

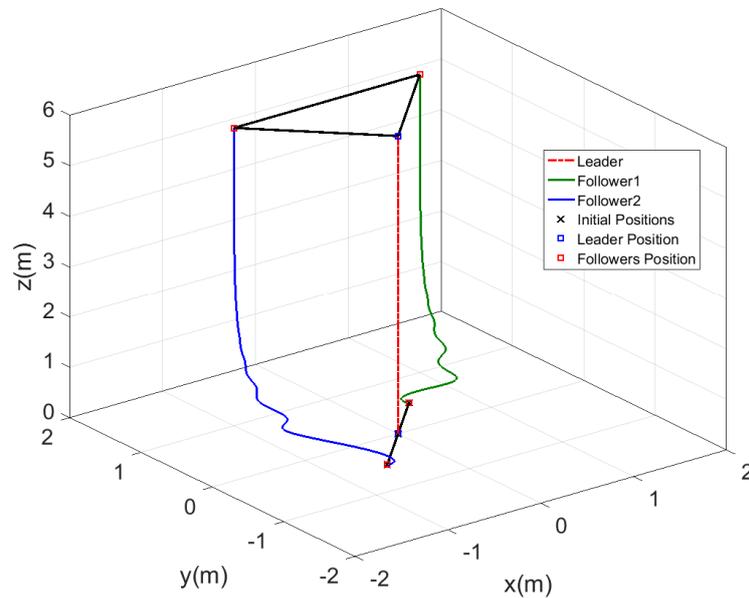


Figure 13. Three dimensional view of fixed triangle formation for MAS.

The reference trajectory signals for the leader agent were a ramp input signal for the x and y position states with starting time = 10 s and slope = 0.01 m, a step input for the altitude z , and a zero signal for the yaw angle ψ , with a full simulation time $t_f = 500$ s.

Figure 14 shows the motion of the leader while tracking the reference trajectory and the followers keeping the required shape while tracking the leader trajectory. The shape was well performed by the three agents using the proposed controllers. The time response for this test is represented in Figure 15, which shows the fast-tracking of the followers to make the required shape with the leader agent without peaks and with a very smooth response.

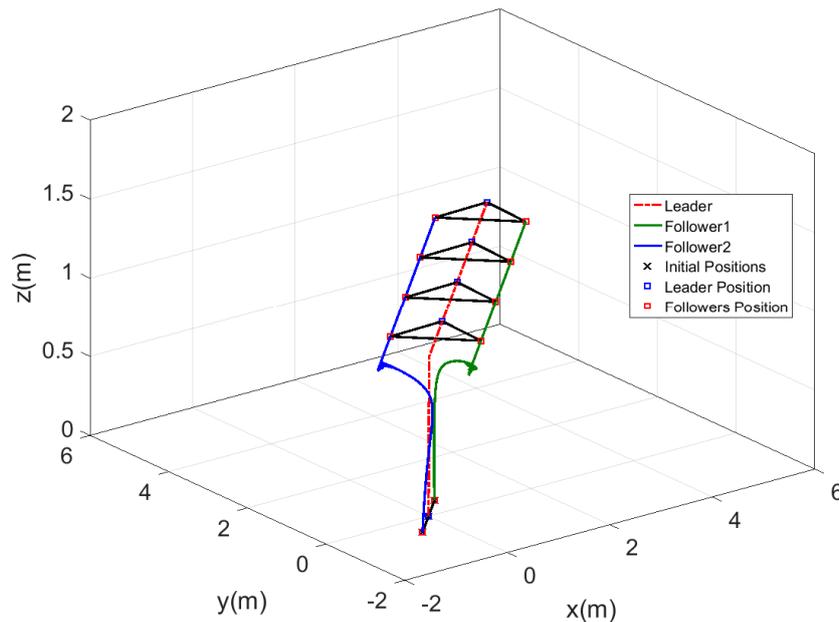


Figure 14. Three-dimensional view of fixed formation tracking ramp trajectory.

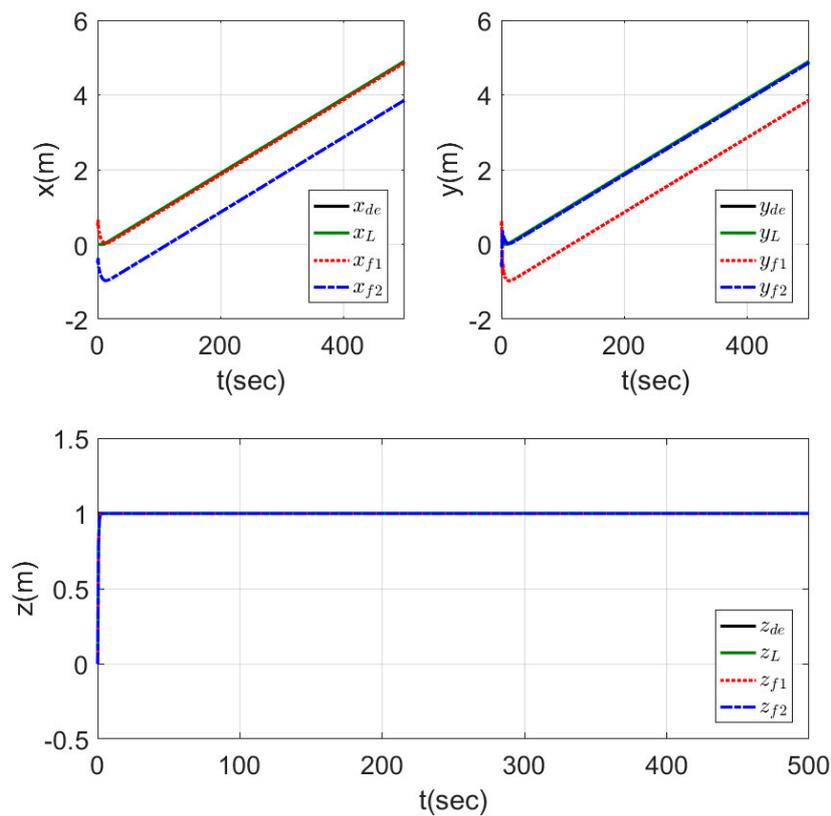


Figure 15. Time response of fixed formation tracking ramp trajectory.

4.4. Leader Orientation-Based Formation Tracking a Helical Trajectory

The previous test showed good trajectory tracking performance while keeping the formation shape fixed. This test differed from the previous one by following the rotation of the leader and keeping the formation shape, but rotation occurred when the leader changed its position. This test is very important for the search and rescue applications of the MAS. In this test, the helical path was used as a reference trajectory for the leader agent. These reference signals are given in Table 10.

Table 10. Leader orientation-based formation tracking a helical trajectory input signals.

State	Reference Trajectory	Time (s)
x	$\cos(0.01\pi t)$	$[25, t_f^*]$
y	$\sin(0.01\pi t)$	$[25, t_f^*]$
z	$0.2 t$	$[0, t_f^*]$
ψ	0	$[0, t_f^*]$

* $t_f = 200$ s.

Another difference in this test as compared to all the cases mentioned previously is that the reference states (ρ_{xi} , ρ_{yi} , ρ_{zi} , and ρ_{ψ_i}) were time-varying here to guarantee that the same triangle shape was kept depending on the leader orientation. The signals of the reference states of the x and y position are given in Table 11, while the altitude and the yaw angle reference states (ρ_{zi} , and ρ_{ψ_i}) were zero for all agents.

Table 11. The x and y reference states.

	Leader	Follower 1	Follower 2
ρ_x	0	$-\cos(\theta_L) + \sin(\theta_L)$	$-\cos(\theta_L) + \sin(\theta_L)$
ρ_y	0	$\cos(\theta_L) - \sin(\theta_L)$	$-\cos(\theta_L) - \sin(\theta_L)$
		$\theta_L = \tan^{-1} \frac{y_L}{x_L}$	

The 3D motion of the agents while forming the triangle shape is shown in Figure 16. The triangle shape was well performed by the agents depending on the orientation of the leader and the path it was following. In Figure 16, each subgraph represents the triangle shape formed by the agents at a different time to show that the orientation of the triangle was changing depending on the leader direction.

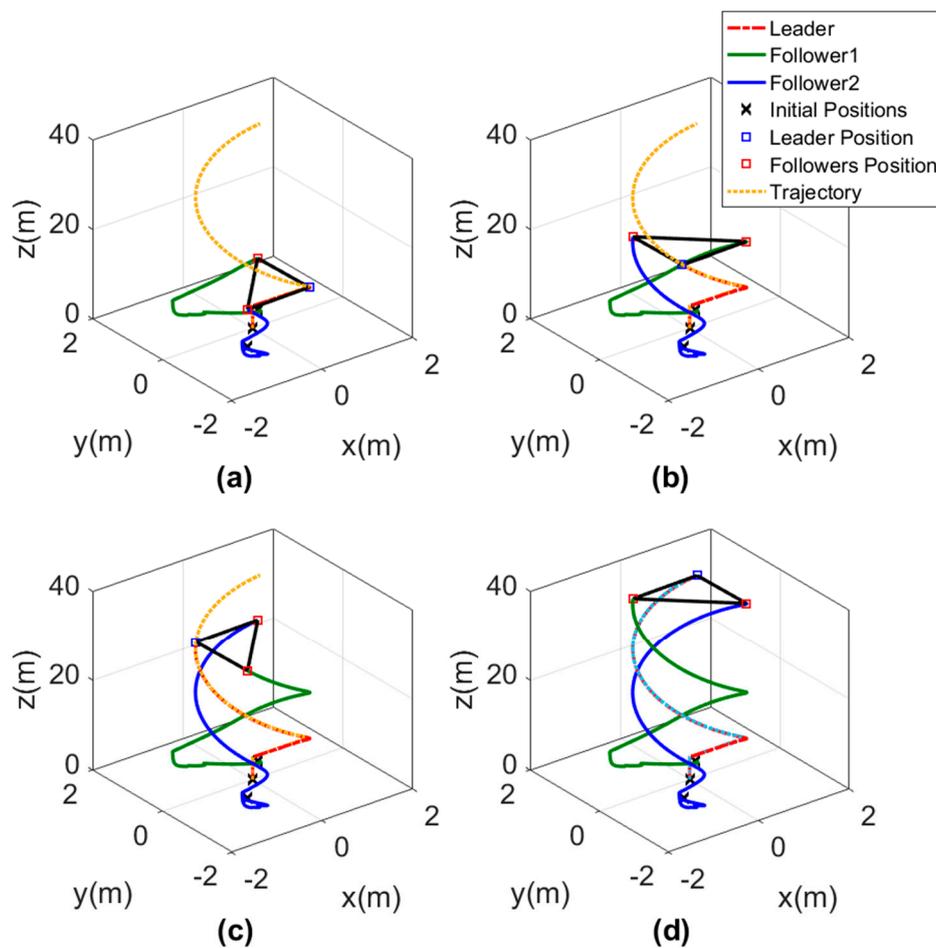


Figure 16. Three-dimensional views of leader orientation-based formation tracking a helical trajectory, (a) Agents position at $t = 50$ s. (b) Agents position at $t = 100$ s. (c) Agents position at $t = 150$ s. (d) Agents position at $t = 200$ s.

The time responses are presented in Figure 17 for the x , y , and z states. The results show a good tracking for the desired signals, while the leader direction was changing the value of the reference states (ρ_x and ρ_y) for the two follower agents (i.e., in second 75 of the simulation $\rho_{xf_1} = 1$, $\rho_{xf_2} = -1$, and $\rho_{yf_1} = \rho_{yf_2} = 1$, which represented a triangle shape directed into the $-ve$ (y) direction, while in second 125 of the simulation $\rho_{xf_1} = \rho_{xf_2} = 1$, and $\rho_{yf_1} = -1$, $\rho_{yf_2} = 1$, which represented a triangle shape with the same dimensions but directed into the $-ve$ (x) direction). The mentioned two examples from the results showed that the shape was kept, while its direction was changed depending on the leader agent.

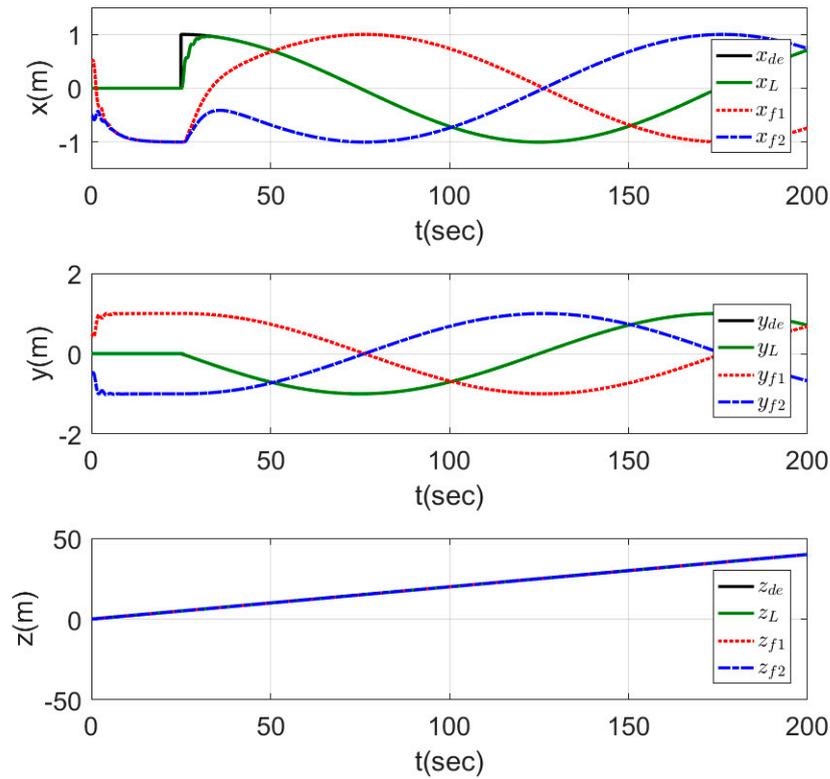


Figure 17. Time response of leader orientation-based formation tracking a helical trajectory.

4.5. Switching Formation Topology

This test required changing the formation shape during hovering mode for the MAS. The reference trajectory signals used for the leader agent are given in Table 12, while the state references for x and y used are given in Table 13.

Table 12. Switching formation topology input signals.

State	Reference Trajectory	Time (s)
x	$-u(t - 50)$	$[110, t_f^*]$
z	$u(t)$	$[0, t_f^*]$
y, ψ	0	$[0, t_f^*]$

* $t_f = 100$ s.

Table 13. The x and y reference states for changing formation.

	Leader	Follower 1	Follower 2
ρ_x	0	$u(t - 25) + u(t - 50)$	$u(t - 25)$
ρ_y	0	$1 - u(t - 50)$	$-1 + u(t - 50) + u(t - 75)$

The results of this test are represented in the plots of Figure 18, which show that the agents formed a line shape on the y -direction in a fast response time through the interval (0 – 25) s of the time. In the second time interval (25 – 50) s, a triangle shape headed by the leader agent and directed to the $-ve$ x -direction was performed by the agents in a smooth and fast response. In the third time interval (50 – 75) s, the leader started to move to the $-ve$ y -direction, while the agents formed a line shape on the x -direction, and the shape was well performed. In the last time interval (75 – 100) s, another

triangle shape was performed but this time was headed by the second follower and was directed to the +ve y -direction.

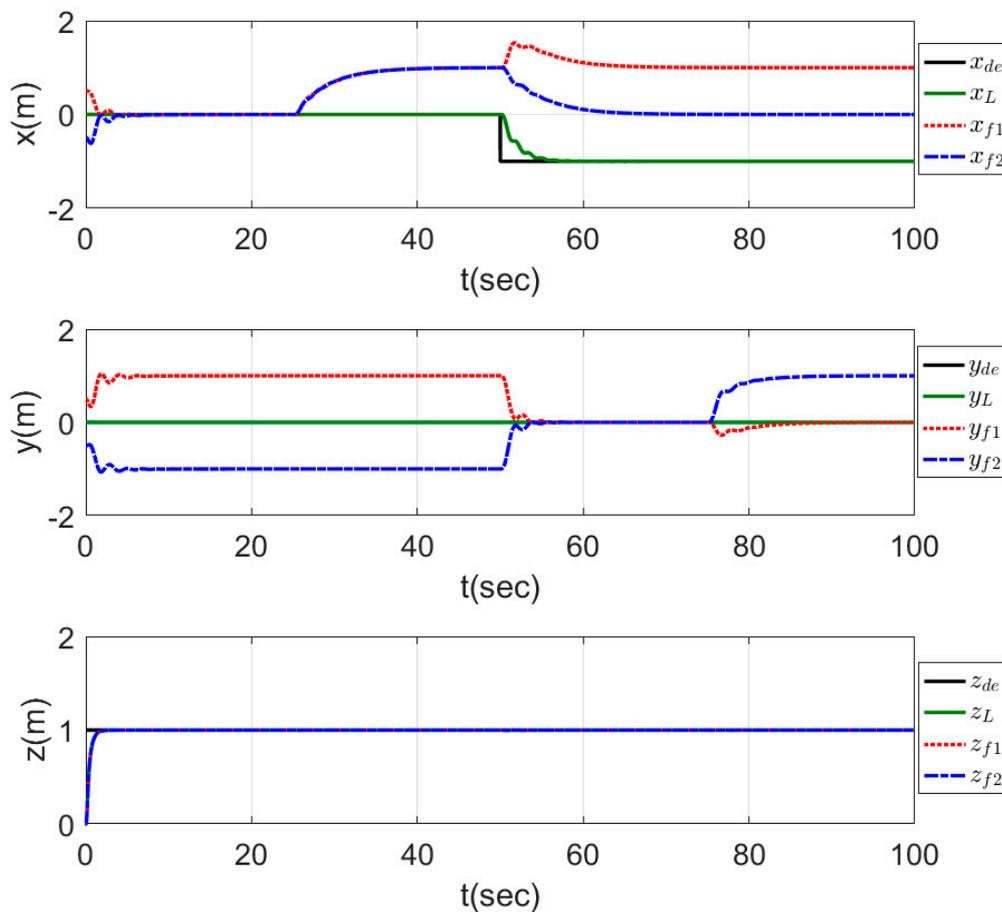


Figure 18. Time response of switching formation topology.

In this test, the 3D motion in Figure 19 was represented in multiple subgraphs to show the exact formation achieved by the MAS. These subgraphs present the motion sequence of the three agents, and it is clear that all the desired formations were performed perfectly without any error.

4.6. Fixed Formation Tracking with Exogenous Disturbances

In this test, the same trajectory and parameters of the fixed formation tracking ramp trajectory test (third test) were used in addition to the existence of different disturbances applied on the leader agent to show the ability of the proposed controller to cope with such practical case. The disturbances applied on the different states of the leader agent were a step of $f_{wz} = 100N$ at $t = 50$ s, pulses for the roll, and pitch and yaw angles ($\tau_{wx} = 1N \cdot m$ at $t = 150$ s, $\tau_{wy} = 1N \cdot m$ at $t = 250$ s, $\tau_{wz} = 1N \cdot m$ at $t = 300$ s) with pulse widths of 50 s.

Figures 20 and 21 show the 3D motion of the agents and the time response of each agent while tracking the trajectory and the disturbances applied. The agents performed the trajectory tracking while different disturbances existed in a perfect way, where small peaks and fast tracking are shown in Figure 21. This proves the high immunity of the proposed controlling scheme against external disturbances.

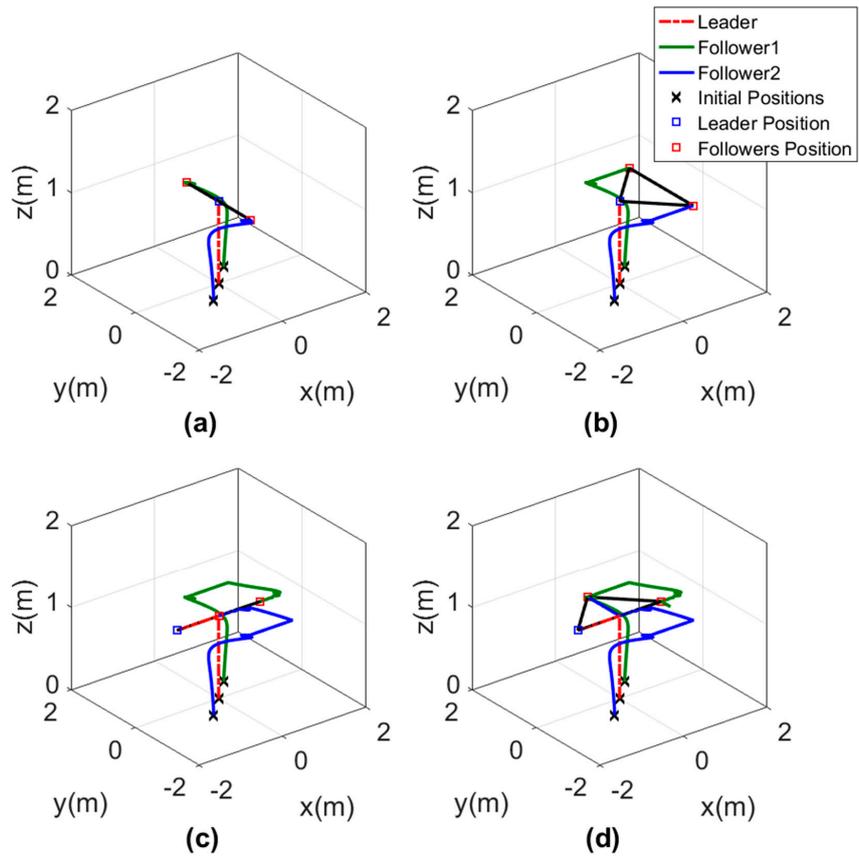


Figure 19. Three-dimensional views of switching formation topology, (a) Agents formation at $t = 25$ s. (b) Agents formation at $t = 50$ s. (c) Agents formation at $t = 75$ s. (d) Agents formation at $t = 100$ s.

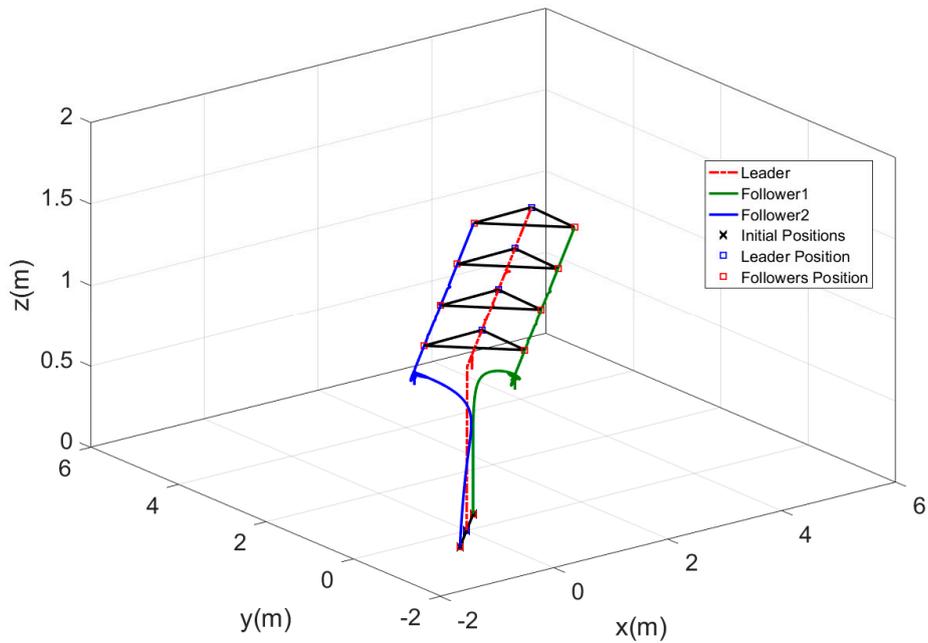


Figure 20. Three-dimensional view of fixed formation tracking with disturbances.

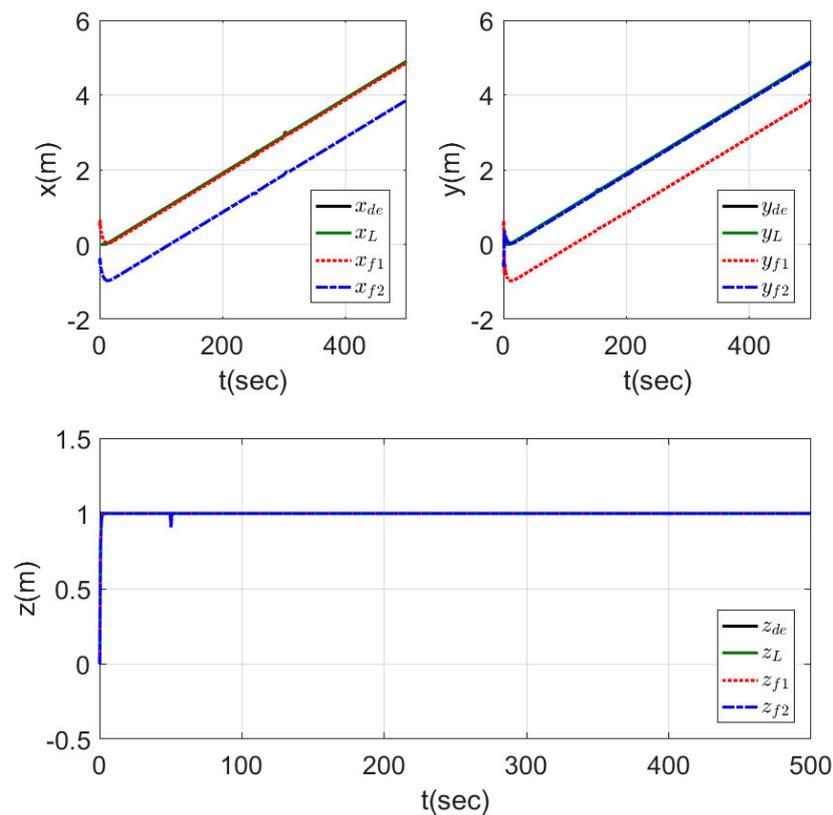


Figure 21. Time response of fixed formation tracking with disturbances.

5. Conclusions and Future Aspects

The consensus control proposed for three agents of quadrotors in this work showed good performance in time. From the simulation cases and the results presented, it can be concluded that the complex nonlinear model for the quadrotor system is treated as a multiple double integrator subsystem. As compared to the other simplified quadrotor models used by some researchers mentioned in the literature survey, the proposed controller outperformed the others by dealing with the complete nonlinear model of the quadrotor system with good performance. This work can be further developed by taking into consideration the following points: proving the closed-loop stability of the proposed control scheme with the nonlinear quadrotor model, developing the proposed controller to achieve containment and flocking control, and implementing the consensus control on a hardware platform using the proposed controlling scheme. Another direction is to extend other control design methods, as in [29,30], to control the single agent UA based on its nonlinear model while using the proposed consensus control scheme.

Author Contributions: Conceptualization, A.A.N. and I.K.I.; Formal analysis, A.A.N., I.K.I. and A.T.A.; Funding acquisition, A.T.A.; Methodology, A.A.N., I.K.I., A.T.A. and A.J.H.; Resources, A.A.N., A.T.A. and A.J.H.; Software, A.A.N., I.K.I. and A.J.H.; Supervision, I.K.I. and A.J.H.; Investigation I.K.I., A.T.A., A.J.H.; Validation, A.T.A. and A.J.H.; Visualization, A.T.A., A.A.N. and A.J.H.; Writing—original draft, A.A.N. and I.K.I.; Writing—review & editing, A.A.N., I.K.I., A.T.A. and A.J.H. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported and funded by Prince Sultan University, Riyadh, Saudi Arabia. We would like to show our gratitude to Prince Sultan University, Riyadh, Saudi Arabia for funding this paper. Special thanks to the Robotics and Internet-of-Things Lab (RIOTU) at the Prince Sultan University, Riyadh, Saudi Arabia.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Appendix A.1. Coordinate Systems

There are two reference frames for the quadrotor system, which are called inertial frame (I) and body frame (K). The inertial frame is a fixed frame (e.g., ground), which means it is fixed relative to the body frame, while the body frame is fixed to the quadrotor system, and it depends on the quadrotor motion [31]. Figure A1 illustrates the two coordinate systems.

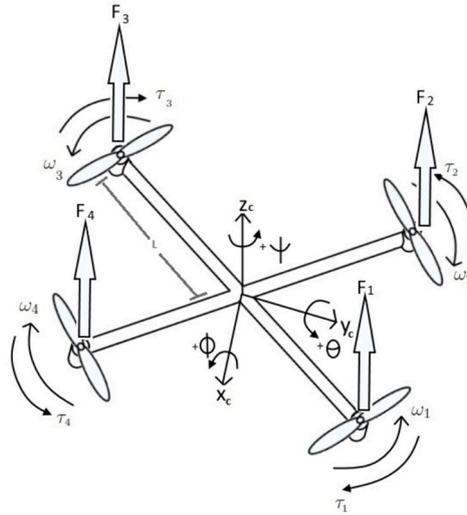


Figure A1. Quadrotor coordinate systems.

Appendix A.2. Euler Angles

Leonhard Euler describes the angular position for a rigid body by using three angles (roll “ ϕ ”, pitch “ θ ”, and yaw “ ψ ”), which are called (Euler angles). These three angles can be given in different ways to describe the angular position in 3D Euclidean space. Euler angles can be used to specify the orientation of a frame relative to another frame, which can be obtained by three successive rotations [31]. In this model, ZYX rotation ($\psi - \theta - \phi$) was used, as shown in Figure A2, and described by the following matrices:

$$\mathbf{R}_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\phi) & -s(\phi) \\ 0 & s(\phi) & c(\phi) \end{bmatrix} \quad (\text{A1})$$

$$\mathbf{R}_y(\theta) = \begin{bmatrix} c(\theta) & 0 & s(\theta) \\ 0 & 1 & 0 \\ -s(\theta) & 0 & c(\theta) \end{bmatrix} \quad (\text{A2})$$

$$\mathbf{R}_z(\psi) = \begin{bmatrix} c(\psi) & -s(\psi) & 0 \\ s(\psi) & c(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{A3})$$

$$\begin{aligned} \mathbf{R}_{zyx}(\phi, \theta, \psi) &= \mathbf{R}_z(\psi) \cdot \mathbf{R}_y(\theta) \cdot \mathbf{R}_x(\phi) \\ &= \begin{bmatrix} c(\theta)c(\psi) & s(\phi)s(\theta)c(\psi) - c(\phi)s(\psi) & c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi) \\ c(\theta)s(\psi) & s(\phi)s(\theta)s(\psi) + c(\phi)c(\psi) & c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi) \\ -s(\theta) & s(\phi)c(\theta) & c(\phi)c(\theta) \end{bmatrix} \end{aligned} \quad (\text{A4})$$

where $c(\) = \cos(\)$, and $s(\) = \sin(\)$.

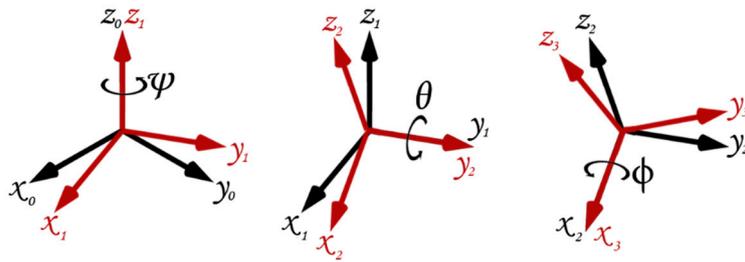


Figure A2. Rotation around Z, Y, and X representation.

Appendix A.3. Model Assumptions

To control a nonlinear, complex, and severely under-actuated system such as a quadrotor system, some assumptions need to be made during the quadrotor modeling process [31]:

- The quadrotor is rigid.
- The quadrotor structure is symmetric.
- The ground effect is neglected.
- The center of gravity origin and the principal axes coincide with the body frame origin and axes.
- Input forces and torques proportional to the squared angular velocity of the rotors.

Appendix A.4. Mathematical Model

The quadrotor nonlinear mathematical model is based on Newton and Euler equations. The two frames (i.e., inertial frame and body frame) are related to each other by the following relations:

$$\begin{cases} \mathcal{V} = \mathbf{R} \cdot \mathcal{V}_B \\ \mathcal{W} = \mathbf{T} \cdot \mathcal{W}_B \end{cases} \quad (\text{A5})$$

where $\mathcal{V} = [x \ y \ z]^T \in \mathbb{R}^3$ is the linear position vector in the earth frame, $\mathcal{W} = [\phi \ \theta \ \psi]^T \in \mathbb{R}^3$ is the angular position vector in the earth frame, $\mathcal{V}_B = [u \ v \ w]^T \in \mathbb{R}^3$ is the linear velocity vector in the body frame, $\mathcal{W}_B = [p \ q \ r]^T \in \mathbb{R}^3$ is the angular velocity vector in the body frame, $\mathbf{R} = \mathbf{R}_{zyx}$, and \mathbf{T} is the angular transformation matrix, which is given by [31]

$$\mathbf{T} = \begin{bmatrix} 1 & s(\phi)t(\theta) & c(\phi)t(\theta) \\ 0 & c(\phi) & -s(\phi) \\ 0 & \frac{s(\phi)}{c(\theta)} & \frac{c(\phi)}{c(\theta)} \end{bmatrix} \quad (\text{A6})$$

where $t(\) = \tan(\)$. The quadrotor kinematic model is given as

$$\begin{cases} \dot{x} = c(\psi)c(\theta)u + [c(\psi)s(\phi)s(\theta) - c(\phi)s(\psi)]v + [s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta)]w \\ \dot{y} = c(\theta)s(\psi)u + [c(\phi)c(\psi) + s(\phi)s(\psi)s(\theta)]v + [c(\phi)s(\psi)s(\theta) - c(\psi)s(\phi)]w \\ \dot{z} = -s(\theta)u + c(\theta)s(\phi)v + c(\phi)c(\theta)w \\ \dot{\phi} = p + s(\phi)t(\theta)q + c(\phi)t(\theta)r \\ \dot{\theta} = c(\phi)q - s(\phi)r \\ \dot{\psi} = \frac{s(\phi)}{c(\theta)}q + \frac{c(\phi)}{c(\theta)}r \end{cases} \quad (\text{A7})$$

The total force applied on the quadrotor system is stated by Newton's law, while the total torque acting on the quadrotor is given by Euler's equation. Both total force and total torque are given by the following relations:

$$\begin{cases} m(\mathcal{W}_B \times \mathcal{V}_B + \dot{\mathcal{V}}_B) = f_B \\ \mathbf{I} \cdot \dot{\mathcal{W}}_B + \mathcal{W}_B \times (\mathbf{I} \cdot \mathcal{W}_B) = m_B \end{cases} \quad (\text{A8})$$

where m is the quadrotor mass, \times is the cross product sign, $\mathbf{f}_B = [f_x \ f_y \ f_z]^T \in \mathbb{R}^3$ is the total force, $\mathbf{m}_B = [m_x \ m_y \ m_z]^T \in \mathbb{R}^3$ is the total torque, and \mathbf{I} is the inertia matrix, which is given as [31]

$$\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \in \mathbb{R}^{3 \times 3}$$

Then, the dynamical model for the quadrotor is given as [31]

$$\begin{cases} f_x = m(\dot{u} + qw - rv) \\ f_y = m(\dot{v} - pw - ru) \\ f_z = m(\dot{w} + pv - qu) \\ m_x = \dot{p}I_{xx} - qrI_{yy} + qrI_{zz} \\ m_y = \dot{q}I_{yy} + prI_{xx} - prI_{zz} \\ m_z = \dot{r}I_{zz} - pqI_{xx} + pqI_{yy} \end{cases} \quad (\text{A9})$$

Appendix A.5. Quadrotor Forces and Moments

Several external forces are applied to the quadrotor and are given here:

$$\mathbf{f}_B = mg\mathbf{R}^T \cdot \hat{\mathbf{k}}_I - f_t \hat{\mathbf{k}}_B + \mathbf{f}_w \quad (\text{A10})$$

where $\hat{\mathbf{k}}_I$ is the inertial z-axis unit vector, $\hat{\mathbf{k}}_B$ is the body z-axis unit vector, g is the gravitational acceleration, f_t is the rotors total thrust, and $\mathbf{f}_w = [f_{wx} \ f_{wy} \ f_{wz}]^T \in \mathbb{R}^3$ are the wind forces applied on the quadrotor system. The external moments applied on the quadrotor system are given as

$$\mathbf{m}_B = \boldsymbol{\tau}_B + \boldsymbol{\tau}_w - \mathbf{m}_g \quad (\text{A11})$$

where $\boldsymbol{\tau}_B = [\tau_x \ \tau_y \ \tau_z]^T \in \mathbb{R}^3$ are the rotor speeds differences torques, $\boldsymbol{\tau}_w = [\tau_{wx} \ \tau_{wy} \ \tau_{wz}]^T \in \mathbb{R}^3$ are the wind torques applied on the quadrotor system, and \mathbf{m}_g is the gyroscopic moments generated by a combination between the four rotors rotation and the quadrotor body, which can be described by the following relation:

$$\mathbf{m}_g = \sum_{i=1}^4 J_p (\boldsymbol{\omega}_B \wedge \hat{\mathbf{k}}_B) (-1)^{i+1} \Omega_i \quad (\text{A12})$$

where J_p is the rotor inertia, which is found to be small enough to be neglected; thus, the gyroscopic moments are excluded from the total equations. Ω_i is the i^{th} rotor angular speed, while the $(-1)^{i+1}$ term represent the direction of rotation for each rotor. Then, the forces and moments are substituted in Equation (A9) to obtain the final dynamical model as follows [31]:

$$\begin{cases} -mg[s(\theta)] + f_{wx} = m(\dot{u} + qw - rv) \\ mg[c(\theta)s(\phi)] + f_{wy} = m(\dot{v} - pw - ru) \\ mg[c(\theta)c(\phi)] + f_{wz} = m(\dot{w} + pv - qu) \\ \tau_x + \tau_{wx} = \dot{p}I_{xx} - qrI_{yy} + qrI_{zz} \\ \tau_y + \tau_{wy} = \dot{q}I_{yy} + prI_{xx} - prI_{zz} \\ \tau_z + \tau_{wz} = \dot{r}I_{zz} - pqI_{xx} + pqI_{yy} \end{cases} \quad (\text{A13})$$

Appendix A.6. Rotor Dynamics Modeling

The inputs for the 6-DOF quadrotor system are combinations of the rotor speed (Ω), which in this case is a force f_t to control the vertical thrust (z) and the torques (τ_x , τ_y , and τ_z) to control the angles (ϕ , θ , and ψ), respectively. Their mathematical relations are given as [31]

$$\begin{cases} f_t = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ \tau_x = bl(\Omega_3^2 - \Omega_1^2) \\ \tau_y = bl(\Omega_4^2 - \Omega_2^2) \\ \tau_z = d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \end{cases} \quad (\text{A14})$$

where b is the thrust coefficient, l is the distance from the quadrotor center to the rotor, and d is the drag coefficient.

The overall nonlinear mathematical model for the quadrotor can be represented as

$$\begin{cases} \dot{x} = c(\psi)c(\theta)u + [c(\psi)s(\phi)s(\theta) - c(\phi)s(\psi)]v + [s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta)]w & (a) \\ \dot{u} = rv - qw - gs(\theta) + \frac{f_{wx}}{m} & (a) \\ \dot{y} = c(\theta)s(\psi)u + [c(\phi)c(\psi) + s(\phi)s(\psi)s(\theta)]v + [c(\phi)s(\psi)s(\theta) - c(\psi)s(\phi)]w & (b) \\ \dot{v} = -ru + pw + gs(\phi)c(\theta) + \frac{f_{wy}}{m} & (b) \\ \dot{z} = -s(\theta)u + c(\theta)s(\phi)v + c(\phi)c(\theta)w & (c) \\ \dot{w} = qu - pv + gc(\theta)c(\phi) + \frac{f_{wz}-f_t}{m} & (c) \\ \dot{\phi} = p + s(\phi)t(\theta)q + c(\phi)t(\theta)r & (d) \\ \dot{p} = \frac{I_{yy}-I_{zz}}{I_{xx}}rq + \frac{\tau_x+\tau_{wx}}{I_{xx}} & (d) \\ \dot{\theta} = c(\phi)q - s(\phi)r & (e) \\ \dot{q} = \frac{I_{zz}-I_{xx}}{I_{yy}}pr + \frac{\tau_y+\tau_{wy}}{I_{yy}} & (e) \\ \dot{\psi} = \frac{s(\phi)}{c(\theta)}q + \frac{c(\phi)}{c(\theta)}r & (f) \\ \dot{r} = \frac{I_{xx}-I_{yy}}{I_{zz}}pq + \frac{\tau_z+\tau_{wz}}{I_{zz}} & (f) \end{cases} \quad (\text{A15})$$

There is another dynamic model for the quadrotor system, which is helpful in the controller design. Using Newton's law, the following is obtained:

$$m\dot{\mathbf{V}} = \mathbf{R}\cdot\mathbf{f}_B = mg\hat{\mathbf{k}}_I - f_t\mathbf{R}\cdot\hat{\mathbf{k}}_B \quad (\text{A16})$$

$$\begin{cases} \ddot{x} = -\frac{f_t}{m}[s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta)] \\ \ddot{y} = -\frac{f_t}{m}[c(\phi)s(\psi)s(\theta) - c(\psi)s(\phi)] \\ \ddot{z} = -\frac{f_t}{m}[c(\phi)c(\theta)] + g \end{cases} \quad (\text{A17})$$

References

1. Ghamry, K.A.; Zhang, Y. Cooperative control of multiple UAVs for forest fire monitoring and detection. In Proceedings of the 12th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA), Auckland, New Zealand, 29–31 August 2016; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2016; pp. 1–6. [CrossRef]
2. Hou, Z.; Wang, W.; Zhang, G.; Han, C. A survey on the formation control of multiple quadrotors. In Proceedings of the 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Jeju, Korea, 28 June–1 July 2017; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2017; pp. 219–225. [CrossRef]
3. Lee, H.; Kim, H.; Kim, H.J. Planning and Control for Collision-Free Cooperative Aerial Transportation. *IEEE Trans. Autom. Sci. Eng.* **2016**, *15*, 189–201. [CrossRef]

4. Xu, B.; Lin, H.; Waghholikar, K.B.; Yang, Z.; Liu, H. Identifying protein complexes with fuzzy machine learning model. *Proteome Sci.* **2013**, *11*, S21. [[CrossRef](#)] [[PubMed](#)]
5. Alkamachi, A.; Erçelebi, E. Modelling and Genetic Algorithm Based-PID Control of H-Shaped Racing Quadcopter. *Arab. J. Sci. Eng.* **2017**, *42*, 2777–2786. [[CrossRef](#)]
6. Ibraheem, I.K.; Ibraheem, G.A. Motion Control of An Autonomous Mobile Robot using Modified Particle Swarm Optimization Based Fractional Order PID Controller. *Eng. Tech. J.* **2016**, *34*, 2406–2419.
7. Ibraheem, I.K.; Ajeil, F.H. Path Planning of an autonomous Mobile Robot using Swarm Based Optimization Techniques. *Al-Khwarizmi Eng. J.* **2017**, *12*, 12–25. [[CrossRef](#)]
8. Ajeil, F.H.; Ibraheem, I.K.; Sahib, M.A.; Humaidi, A.J. Multi-objective path planning of an autonomous mobile robot using hybrid PSO-MFB optimization algorithm. *Appl. Soft Comput.* **2020**, *89*, 106076. [[CrossRef](#)]
9. Abbas, N.H.; Sami, A.R. Tuning of PID Controllers for Quadcopter System using Hybrid Memory based Gravitational Search Algorithm–Particle Swarm Optimization. *Int. J. Comput. Appl. Technol.* **2017**, *172*, 9–18.
10. Wooldridge, M.J. *An Introduction to Multiagent Systems*, 2nd ed.; John Wiley & Sons: Chichester, UK, 2009.
11. Chong, C.-Y. Forty years of distributed estimation: A review of noteworthy developments. In Proceedings of the 2017 Sensor Data Fusion: Trends, Solutions, Applications (SDF), Bonn, Germany, 10–12 October 2017; 2017; pp. 1–10. [[CrossRef](#)]
12. Li, Z.; Duan, Z. *Cooperative Control of Multi-Agent Systems: A Consensus Region Approach*; CRC Press: Florida, FL, USA, 2017.
13. Ren, W.; Beard, R.W. *Distributed Consensus in Multi-vehicle Cooperative Control*; Springer: London, UK, 2008.
14. Mesbahi, M.; Egerstedt, M. *Graph Theoretic Methods in Multiagent Networks*; Princeton University Press: New Jersey, NJ, USA, 2010.
15. Chen, T.; Shan, J. Distributed Tracking of a Class of Underactuated Lagrangian Systems With Uncertain Parameters and Actuator Faults. *IEEE Trans. Ind. Electron.* **2020**, *67*, 4244–4253. [[CrossRef](#)]
16. Wang, Y.; Wu, Q.; Wang, Y.; Yu, D. Consensus algorithm for multiple quadrotor systems under fixed and switching topologies. *J. Syst. Eng. Electron.* **2013**, *24*, 818–827. [[CrossRef](#)]
17. Eskandarpour, A.; Majd, V.J. Cooperative formation control of quadrotors with obstacle avoidance and self collisions based on a hierarchical MPC approach. In Proceedings of the Second RSI/ISM International Conference on Robotics and Mechatronics (ICRoM), Tehran, Iran, 15–17 October 2014; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2014; pp. 351–356. [[CrossRef](#)]
18. Rabah, A.; Qing-he, W. Adaptive Leader Follower Control for Multiple Quadrotors via Multiple Surfaces Control. *J. Beijing Inst. Technol.* **2016**, *25*, 526–532.
19. Alrawi, W.M.J. Improving Leader-Follower Formation Control Performance for Quadrotors. Ph.D. Thesis, University of Essex, Colchester, UK, 2016.
20. Huang, Z. Consensus Control of Multiple-Quadcopter Systems Under Communication Delays. Master's Thesis, Dalhousie University, Halifax, NS, Canada, 2017.
21. Mahfouz, M.; Hafez, A.T.; Ashry, M.; Elnashar, G. Formation Configuration for Cooperative Multiple UAV Via Backstepping PID Controller. 2018. Available online: <https://arc.aiaa.org/doi/abs/10.2514/6.2018-5282> (accessed on 16 June 2020).
22. Du, H.; Zhu, W.; Wen, G.; Duan, Z.; Lu, J. Distributed Formation Control of Multiple Quadrotor Aircraft Based on Nonsmooth Consensus Algorithms. *IEEE Trans. Cybern.* **2019**, *49*, 342–353. [[CrossRef](#)] [[PubMed](#)]
23. Najm, A.A.; Ibraheem, I.K. Nonlinear PID controller design for a 6-DOF UAV quadrotor system. *Eng. Sci. Technol. Int. J.* **2019**, *22*, 1087–1097. [[CrossRef](#)]
24. Najm, A.A.; Ibraheem, I.K. Altitude and Attitude Stabilization of UAV Quadrotor System using Improved Active Disturbance Rejection Control. *Arab. J. Sci. Eng.* **2020**, *45*, 1985–1999. [[CrossRef](#)]
25. Isira, A.S.M. Consensus Control of a Class of Nonlinear Systems. Ph.D. Thesis, University of Manchester, Manchester, UK, 2016.
26. Riyadh, W.; Kasim, I.; Abdul-Adheem, W.R.; Ibraheem, I.K. From PID to Nonlinear State Error Feedback Controller. *Int. J. Adv. Comput. Sci. Appl.* **2017**, *8*, 312–322. [[CrossRef](#)]
27. Kasim, I.; Riyadh, W. On the Improved Nonlinear Tracking Differentiator based Nonlinear PID Controller Design. *Int. J. Adv. Comput. Sci. Appl.* **2016**, *7*, 234–241. [[CrossRef](#)]
28. Ibraheem, I.K. Anti-Disturbance Compensator Design for Unmanned Aerial Vehicle. *J. Eng.* **2019**, *26*, 86–103. [[CrossRef](#)]

29. Abdul-Adheem, W.R.; Ibraheem, I.K. An Improved Active Disturbance Rejection Control for a Differential Drive Mobile Robot with Mismatched Disturbances and Uncertainties. Available online: <https://arxiv.org/abs/1805.12170> (accessed on 25 May 2018).
30. Humaidi, A.J.; Ibraheem, I.K. Speed Control of Permanent Magnet DC Motor with Friction and Measurement Noise Using Novel Nonlinear Extended State Observer-Based Anti-Disturbance Control. *Energies* **2019**, *12*, 1651. [[CrossRef](#)]
31. Sabatino, F. Quadrotor Control: Modeling, Nonlinear Control Design, and Simulation. Master's Thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2015.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).