AUT Journal of Electrical Engineering
10.22060/EEJ.2025.23822.5635

Formation Tracking of Nonlinear Quadcopters using Robust Model Predictive Control in Unknown Environments

Navid Mohammadi[1] Saeed Khankalantary[*2] Morteza Tayefi[3]

[1,3]Institute of Intelligent Control Systems, K. N. Toosi University of Technology, Tehran, Iran

[2]Faculty of Electrical Engineering, K. N. Toosi University of Technology, Tehran, Iran

Corresponding author, s.kalantary@kntu.ac.ir

Abstract:

This study presents a robust framework for formation tracking and cooperative control of multiple nonlinear quadcopters, incorporating actuator dynamics, wind drag, and uncertainties. Instead of using torques and forces as control inputs, the theoretical models are developed based on motor speed controls. Designing a controller directly based on actuator signals minimizes uncertainty, improves real-world applicability, and eliminates the need for an additional control allocation algorithm. Two optimal controllers, Linear Quadratic Regulator and Model Predictive Control, are developed and evaluated under dynamic conditions, including uncertainty, disturbances, and obstacle-laden environments. The error-based Predictive Control design minimizes sensitivity to initial conditions and enhances robustness, outperforming Linear Quadratic Regulator in trajectory tracking, energy efficiency, and stability. This framework is extended to a graph-theoretic multi-agent system with four interconnected quadcopters, employing formation control to navigate complex helical paths while maintaining square formations among uncertainties and obstacles. Integrating advanced control methods and graph theory demonstrates coordination and robustness, validating the system's potential for surveillance, rescue missions, and autonomous transfer applications. This research lays the groundwork for extending these methods to larger, heterogeneous agent teams, which can ensure adaptability and precision in real-world scenarios.

Keywords:

Formation Tracking, Quadcopter, Nonlinear Dynamics, Multi-Agent Control, Model Predictive Control

1. Introduction

Quadcopters have recently emerged as highly versatile and stable Unmanned Aerial Vehicles (UAVs), exhibiting immense potential for autonomous control. Their applications span a wide spectrum, encompassing surveillance, inspection, search, rescue missions, and transportation, positioning them at the forefront of modern technological advancements. Coordinating multiple quadcopters holds promise for extending their capabilities and adaptiveness in complex environments. This article encompasses a detailed investigation of control strategies, formation modeling, and guidance of multiple quadcopters in synchronized flight. In addition to providing the necessary research background, it aims to establish a robust foundation for developing distributed coordination algorithms, enabling quadcopters to tackle complex tasks collaboratively.

A crucial aspect of multi-agent systems is the precise modeling and control of individual quadcopters. Unlike many existing approaches that rely on simplified dynamic models, our work employs a comprehensive Newton–Euler-based model that explicitly incorporates actuator dynamics, rotor delays, and aerodynamic effects—thereby ensuring a more realistic and robust control of each quadcopter. The work by [1] establishes a dynamic model for quadcopters, subsequently evaluating the effectiveness of Proportional – Integral – Derivative (PID) and sliding mode controllers for various tasks, ranging from hovering to precise trajectory tracking. Reference [2] provides a comprehensive tutorial on quadcopter configurations and explores the intricacies of six degrees of freedom modeling. It also presents methods for tuning PID for altitude, position, and speed control. Nonlinear control techniques like sliding mode and back-stepping for stabilizing attitude and position are compared in [3].

Consensus protocols are of paramount importance in the coordination of multiple agents, which facilitate decentralized decision-making. In this context, [4] establishes the stability of a consensus approach for velocity synchronization and formation control within second-order multi-agent systems. Reference [5] introduces the utilization of graph Laplacians to delineate formation shapes based on agent connectivity, with a distributed linear control law proven to stabilize shapes residing in the null space of the graph. Furthermore, the research by [6] demonstrates that an error predictor-based control law achieves formation tracking in fractional-order multi-agent systems under a directed graph with a spanning tree, validated through theory and simulations.

Formation flight emerges as a pivotal application within the domain of multi-quadcopter systems. [7] conducts an exhaustive review of control strategies, encompassing leader-follower dynamics, artificial potentials, and rigid formations. Subsequently, [8] establishes conditions for graph controllability for leader-follower nearest neighbor formations while [9] formulates theorems regarding formation maintenance, task duration, and velocity for robots leveraging control Lyapunov functions. Additionally, [10] presents an Model Predictive Control (MPC)-based controller for maintaining formation stability under variable communication time delays, ensuring collision-free motion along desired paths. Arbitrary formations using bijective coordinate transformations of regular polygons are generated in [11]. While existing works address formation flight through strategies such as leader-follower dynamics and MPC under communication delays, by considering detailed nonlinear dynamics and graph-based formation rules, we develop our approach to achieve stability and trajectory tracking in complex environments.

Integrating potential field navigation with leader-follower control, as outlined in [12], manifests as a powerful tool for achieving robust quadcopter formations, even in cluttered environments. Similarly, [13] introduces a robust formation control approach leveraging MPC, which incorporates virtual structures and artificial potential fields to ensure trajectory tracking and obstacle avoidance. Recent studies have further advanced formation control strategies. For instance, [14] proposes a distributed MPC framework that uses both soft and hard constraints to ensure collision avoidance and maintain communication connectivity, with its effectiveness demonstrated through real experiments. [15] integrates MPC with an extended state observer to compensate for external disturbances and internal uncertainties, leading to precise trajectory tracking in complex environments. Additionally, [16] employs geometric control methods based on the SE(2) framework to design robust control laws for stabilization and formation tracking in nonholonomic systems, which are applied to leader-follower configurations. In contrast to recent studies that utilize potential field navigation and geometric control for formation tracking, our framework integrates these elements within an error-based MPC strategy, resulting in improved transient performance and effective handling of disturbances and actuator constraints. Reference [17] develops a 12-state model and

10.22060/EEJ.2025.23822.5635

implements MPC only on the roll and pitch loops—leaving the remaining translational states under PD control—and validates performance under nominal conditions without explicitly modeling uncertainty or robustness. In contrast, our work employs a full nonlinear quadcopter model, applies error-based MPC uniformly across the entire state vector, and rigorously demonstrates convergence and robustness under uncertainty, external disturbances, obstacles, and actuator faults.

This research advances quadcopters' modeling, control, and coordination under nonlinear, uncertain, and dynamic conditions, presenting a robust framework for real-world applications. A comprehensive nonlinear dynamics model is developed, incorporating wind drag effects, actuator dynamics, and the inherent time delays of the actuators. These developments ensure a realistic and detailed representation of the quadcopter's unstable behavior, surpassing conventional modeling approaches. Existing methods usually linearize quadcopter dynamics, assuming torques and forces as control inputs, and then employ control allocation for real motor speeds. This work directly integrates actuator dynamics and delay into the linearized structure, allowing for seamless and accurate controller design.

In this paper, the MPC strategy utilizes error dynamics instead of state variables to enhance robustness by reducing sensitivity to initial conditions. The MPC controller is compared against the Linear Quadratic Regulator (LQR) controller for stabilization and trajectory tracking tasks under nonlinear conditions, including disturbances and obstacles. Results demonstrate the superior performance of the MPC controller, offering smoother and more precise control signals and maintaining stability in the presence of transient uncertainties. Extending the work to multi-agent systems, the research employs graph-based formation control for four interconnected quadcopters, modeling them as nodes within a graph structure. Formation rules are developed for a square configuration, with a leader quadcopter assigned a complex helical path. The follower agents maintain formation geometry by relying on graph connectivity, ensuring robust and collision-free operation despite environmental disturbances and uncertainties. This framework integrates graph theory and formation control with advanced path planning, enabling the leader-follower system to navigate intricate trajectories while preserving inter-agent connectivity and stability. By incorporating wind drag effects, actuator delays, and modeling uncertainties directly into the nonlinear dynamics and the control designs, this research achieves an important level of fidelity and robustness, addressing critical challenges in real-world quadcopter systems and advancing applications in areas such as surveillance, autonomous transportation, and rescue missions.

2. Quadcopter mathematical modeling

The nonlinear model governing quadcopter kinematics, rigid body dynamics, actuator dynamics, and subsequent linearization are discussed in this section. We thoroughly analyze the system dynamics using small angle approximations for roll and pitch motions to enable control system design around its equilibrium. The actuator dynamics employs a first-order transfer function to faithfully represent the relationship between rotor speed set points and actual values.

To represent body-to-inertial frame vector transformations, we leverage a rotation matrix based on Z-Y-X Euler angles, as elaborated in [18] and depicted in Fig. 1.

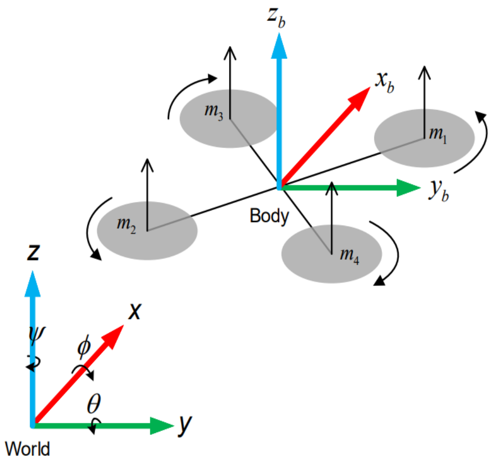


Fig 1. Quadrotor coordinate diagram

After thoroughly examining the underlying assumptions and rotation matrices, we employ the Newton-Euler equations to derive a set of nonlinear equations. According to [19], [20], the nonlinear state equations of the quadcopter are developed as

$$\dot{\mathbf{X}}=\begin{cases}\dot{x}=v_x\\ \dot{y}=v_y\\ \dot{z}=v_z\\ \dot{v}_x=\dfrac{T_{mc}}{m}\left(cos\left(\varphi\right)sin\left(\theta\right)cos\left(\psi\right)+sin\left(\varphi\right)sin\left(\psi\right)\right)-A_x\\ \dot{v}_y=\dfrac{T_{mc}}{m}\left(sin\left(\psi\right)sin\left(\theta\right)cos\left(\varphi\right)-cos(\psi)sin(\varphi)\right)-A_y\\ \dot{v}_z=\dfrac{T_{mc}}{m}\left(cos(\varphi)cos(\theta)\right)-A_z-g\\ \dot{\varphi}=p+sin\left(\varphi\right)tan\left(\theta\right)q-cos\left(\varphi\right)tan\left(\theta\right)r\\ \dot{\theta}=qcos\left(\varphi\right)+rsin(\varphi)\\ \dot{\psi}=-\dfrac{sin\left(\varphi\right)}{cos\left(\theta\right)}q+\dfrac{cos\left(\varphi\right)}{cos\left(\theta\right)}r\\ \dot{p}=\dfrac{I_y-I_z}{I_x}qr+\dfrac{\tau_x}{I_x}\\ \dot{q}=\dfrac{I_z-I_x}{I_y}pr+\dfrac{\tau_y}{I_y}\\ \dot{r}=\dfrac{I_x-I_y}{I_z}pq+\dfrac{\tau_z}{I_z}\end{cases} \tag{1}$$

where x,y,z denote positions, v_x,v_y,v_z represent velocities, $I_{x,}I_{y,}I_z$ denote moments of inertia along the x,y,z. The Euler angles φ,θ,ψ represent roll, pitch, and yaw angles, respectively. g denotes the acceleration due to gravity, T_{mc} signifies multirotor thrust and τ_x, τ_y , τ_z stand for multirotor torques around the x,y,z axes. $A_x=0.5\rho\bar{v}_x^2\mathrm{sc}_d$, $A_y=0.5\rho\bar{v}_y^2\mathrm{sc}_d$, $A_z=0.5\rho\bar{v}_z^2\mathrm{sc}_d$ are aerodynamic drag forces. $\bar{\mathbf{v}}_{\mathbf{x,y,z}}=\mathbf{v}+\mathbf{v}_{\mathbf{w}}$ and $\mathbf{v}_{\mathbf{w}}$ is the wind velocity in three directions x,y,z. The quadcopter thrust and torque with respect to motor speeds are formulated as:

$$T_{mc}=k_f(\omega_1^2+\omega_2^2+\omega_3^2+\omega_4^2)$$

$$\tau_x=k_f l(\omega_2^2-\omega_4^2) \tag{2}$$

$$\tau_y = k_f l(\omega_3^2 - \omega_1^2)$$

$$\tau_z = k_m(\omega_1^2 + \omega_3^2 - \omega_2^2 - \omega_4^2)$$

where l is the distance between two rotors and the coefficients k_f and k_m are determined based on the specifications of the propellers employed in the quadcopter configuration.

The motor speed response in a quadcopter control system is characterized by the following transfer function from throttle command to motor speed [21]:

$$\Delta\omega_i = \frac{1}{T_m s + 1}\Delta\omega_i e^{-\sigma s} \tag{3}$$

Equation (3) describes the relationship between the throttle command and the resulting motor speed (ω_i) in a quadcopter control system. The transfer function involves a first-order system with a time constant (T_m) and delay factor (σ). The term $\frac{1}{T_m s + 1}$ represents the first-order dynamics and $e^{-\sigma s}$ introduces the delay. This equation concisely represents how the motor speed responds to changes in the throttle command, considering both the inherent dynamics (T_m) and any system delay (σ).

The torque τ can be calculated from the following equation:

$$\tau = \begin{bmatrix} l(f_2 - f_4) \\ l(f_3 - f_1) \\ \sum M_i \end{bmatrix} \tag{4}$$

where $f_i = k_f \omega_i^2$, $M_i = \frac{k_m}{k_f} f_i = \mu f_i$.

For the conversion of input torques into motor forces, one may employ the following matrix relationship.

$$\begin{bmatrix} T_{mc} \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & l & 0 & -l \\ -l & 0 & 0 & 0 \\ \mu & -\mu & \mu & -\mu \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \tag{5}$$

The torque equation can also be extended as

$$\begin{aligned} \tau_x &= k_f l(\omega_2^2 - \omega_4^2) \\ \tau_y &= k_f l(\omega_3^2 - \omega_1^2) \\ \tau_z &= k_m(\omega_1^2 + \omega_3^2 - \omega_2^2 - \omega_4^2) \end{aligned}$$

Now, by expressing motor speeds as $\omega_i = \omega_0 + \Delta\omega_i$, we rewrite the equations. w_0 represents the motor speed in the equilibrium state or quadcopter hover so

$$mg = 4k_f\omega_0^2 \rightarrow \omega_0 = \frac{1}{2}\sqrt{\frac{mg}{k_f}}$$

and $\Delta\omega_i$ is the speed deviation from the equilibrium value. Let

$$\omega_i^2 = \omega_0 + 2\omega_0\Delta\omega_i + \Delta\omega_i{}^2$$

Hence, for small deviations around values ($\Delta\omega_i \approx 0$), it is given by

$$\omega_i^2 = \omega_0 + 2\omega_0\Delta\omega_i$$

Quadcopter motion is approximately modeled by linearized equations, assuming slight deviations from the hover state. When the angles are small $((sin(\theta) \approx sin(\varphi) \approx 0$ and $\bar{\mathbf{v}}_{\mathbf{x,y,z}} = \mathbf{v} + \mathbf{v}_{\mathbf{w}}$) and assuming zero yaw, the translational equations for the x, y, z axes simplify to

$$\begin{aligned}
&\dot{v}_x = g\theta \\
&\dot{v}_y = -g\varphi \\
&\dot{v}_z = \frac{T_{mc}}{m} - g = \frac{T_{mc_{new}}}{m} = \frac{2k_f\omega_0}{m}\Delta\omega_i \rightarrow Tmc = T_{mc_{new}} + mg
\end{aligned} \tag{6}$$

Because we linearize about hover with negligible velocities, we omit aerodynamic drag in equation (6). where thrust equals the weight ($T_{mc} = mg$). When the angles are small, assuming $qr \approx pr \approx pq \approx 0$, the angular equations are simplified as follows:

$$\begin{aligned}
&\dot{p} = \ddot{\varphi} = \frac{\tau_x}{I_x} = \frac{2lk_f\omega_0}{I_x}(\Delta\omega_2 - \Delta\omega_4) \\
&\dot{q} = \ddot{\theta} = \frac{\tau_y}{I_y} = \frac{2lk_f\omega_0}{I_y}(\Delta\omega_3 - \Delta\omega_1) \\
&\dot{r} = \ddot{\psi} = \frac{\tau_z}{I_z} = \frac{2k_m\omega_0}{I_z}(\Delta\omega_1 - \Delta\omega_2 + \Delta\omega_3 - \Delta\omega_4)
\end{aligned} \tag{7}$$

Then, the linearized dynamics are described in the state space form as

$$\begin{aligned}
&\dot{\mathbf{X}} = \mathbf{AX}(t) + \mathbf{Bu}(t) \\
&\mathbf{y} = \mathbf{CX}(t)
\end{aligned} \tag{8}$$

where $\mathbf{X}(t) = \begin{bmatrix} \varphi\ \dot{\varphi}\ \theta\ \dot{\theta}\ \Psi\ \dot{\Psi}\ x\ v_x\ y\ v_y\ z\ v_z \end{bmatrix}^T$ and $\mathbf{u}(t) = [\Delta\omega_1, \Delta\omega_2, \Delta\omega_3, \Delta\omega_4]^T$. According to equations (2), (5) and (7) state matrices are defined as follows:

$$\mathbf{A}=\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{B}=\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{2lk_f\omega_0}{I_x} & 0 & -\frac{2lk_f\omega_0}{I_x} \\ 0 & 0 & 0 & 0 \\ -\frac{2lk_f\omega_0}{I_y} & 0 & \frac{2lk_f\omega_0}{I_y} & 0 \\ 0 & 0 & 0 & 0 \\ \frac{2k_m\omega_0}{I_z} & -\frac{2k_m\omega_0}{I_z} & \frac{2k_m\omega_0}{I_z} & -\frac{2k_m\omega_0}{I_z} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{2k_f\omega_0}{m} & \frac{2k_f\omega_0}{m} & \frac{2k_f\omega_0}{m} & \frac{2k_f\omega_0}{m} \end{bmatrix}$$

$$\mathbf{C}=diag\left(\left[1,1,1,1,1,1,1,1,1,1,1,1\right]\right)$$

$$\mathbf{D}=\bar{0}_{12*4}$$

This linear model is utilized to devise LQR and MPC control strategies for the nonlinear quadcopter.

3. Quadcopter control frameworks

3-1- LQR controller

The LQR control is designed to optimize the performance of a linear time-invariant system by minimizing a quadratic cost function while ensuring system stability. The fundamental idea behind LQR is to find an optimal state feedback control law that minimizes a weighted sum of the state and control efforts. The performance of the system is evaluated using a quadratic cost function of the form [22]:

$$J=\frac{1}{2}\int_{t_0}^{t_f}[(\mathbf{X}(t)-\mathbf{X}_d(t))^T\mathbf{Q}(\mathbf{X}(t)-\mathbf{X}_d(t))+\mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t)]dt \tag{9}$$

In equation (9), $\mathbf{Q}$ and $\mathbf{R}$ are special matrices showing the importance of the control inputs and the states in the cost function. $\mathbf{X_d}$ is the desired state that guides the quadcopters to be in the right formation, following some rules. The goal of making an LQR controller is to create a rule for controlling the state like this:

$$\mathbf{u}(t)=-\mathbf{K}(\mathbf{X}(t)-\mathbf{X}_d(t)) \tag{10}$$

where $\mathbf{K}$ is the gain matrix that minimizes the cost function J and ensures stability. The gains $\mathbf{K}$ can be computed by solving the continuous-time algebraic Riccati differential equation:

$$\mathbf{A}^T\mathbf{P}+\mathbf{P}\mathbf{A}-\mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}+\mathbf{Q}=0 \tag{11}$$

where $\mathbf{P}$ is the solution to the Riccati equation. Once $\mathbf{P}$ is obtained, the optimal state feedback gains $\mathbf{K}$ can be calculated as $\mathbf{K}=\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}$.

3-2- MPC controller

MPC is a sophisticated control strategy that has gained significant popularity in recent decades. Unlike traditional control methods, MPC is a receding horizon control approach that solves an optimization problem at each time step to determine the control input. It provides a powerful framework for controlling a wide range of dynamic systems, including those that are nonlinear, time-varying, and subject to constraints.

The core concept of MPC involves forecasting the system's future actions within a limited prediction horizon and fine-tuning control inputs to minimize a predefined cost. By looking ahead, MPC considers future states and adapts control inputs accordingly. At each time step, the optimization problem is solved, and only the initial control input from the obtained sequence is implemented. This cycle repeats at each step, forming a dynamic horizon of control choices.

The typical formulation of the MPC optimization problem is as follows [23]:

$$\begin{aligned}&J=\mathbf{X}^{T}\left(k+N_p\,|\,k\right)\mathbf{P}\mathbf{X}\left(k+N_p\,|\,k\right)+\\&\sum_{i=1}^{N_p\text{-}1}\left(\mathbf{X}\left(k+i\,|\,k\right)\text{-}\mathbf{X}_d\left(k+i\,|\,k\right)\right)^{T}\mathbf{Q}(\mathbf{X}\left(k+i\,|\,k\right)\text{-}\mathbf{X}_d\left(k+i\,|\,k\right))+\\&\sum_{j=1}^{N_c}\mathbf{U}^{T}\left(k+j\text{-}1\,|\,k\right)\mathbf{R}\mathbf{U}\left(k+j\text{-}1\,|\,k\right)\end{aligned}\tag{12}$$

The prediction equations for a system model can be represented as:

$$\begin{aligned}\mathbf{X}\left(k+i+1\,|\,k\right)&=\mathbf{A}\mathbf{X}\left(k\,|\,k\right)+\mathbf{B}\mathbf{u}\left(k\,|\,k\right)\\\mathbf{y}\left(k+i\,|\,k\right)&=\mathbf{C}\mathbf{X}\left(k\,|\,k\right)+\mathbf{D}\mathbf{u}\left(k\,|\,k\right)\end{aligned}\tag{13}$$

According to equation (13), as well as the relationship between the state and control variables $\mathbf{X}\left(k\,|\,k\right)$, $\mathbf{u}\left(k\,|\,k\right)$ the prediction of the state $\mathbf{X}\left(k+1\,|\,k\right)$ can be obtained as

$$\begin{aligned}&\mathbf{X}_{n\times1}\left(k+1\,|\,k\right)=\mathbf{A}_{n\times n}\mathbf{X}_{n\times1}\left(k\,|\,k\right)+\mathbf{B}_{n\times m}\mathbf{u}_{m\times1}\left(k\,|\,k\right)\\&\mathbf{X}\left(k+2\,|\,k\right)=\mathbf{A}^{2}\mathbf{X}\left(k\,|\,k\right)+\mathbf{A}\mathbf{B}\mathbf{u}\left(k\,|\,k\right)+\mathbf{B}\mathbf{u}_{c}\left(k+1\,|\,k\right)\\&\vdots\\&\mathbf{X}\left(k+N_p\,|\,k\right)=\mathbf{A}^{N_p}\mathbf{X}\left(k\,|\,k\right)+\mathbf{A}^{N_p\text{-}1}.\mathbf{B}.\mathbf{u}\left(k\,|\,k\right)+\ldots+\mathbf{A}^{N_p\text{-}N_c}.\mathbf{B}.\mathbf{u}\left(k+N_c\text{-}1\,|\,k\right)\end{aligned}\tag{14}$$

In this paper, the MPC prediction horizon represents the future time window over which the system's behavior is forecasted based on the current state and model dynamics. This horizon, denoted by N_p, allows the controller to anticipate future disturbances and system evolutions, thereby enabling it to plan a sequence of control actions that drive the system toward the desired trajectory. In contrast, the control horizon, denoted by N_c, defines the interval over which the control inputs are actively optimized. Beyond this point, the control signals are usually held constant or follow a pre-established pattern to simplify computations. By carefully selecting the lengths of these horizons, one can balance the trade-off between computational complexity and control performance—longer prediction horizons may provide a more comprehensive outlook on future states but require more computational resources, whereas shorter control horizons help in reducing the optimization burden while still ensuring effective control over critical future periods.

Let

$$\mathbf{X}_{nN_p\times 1} = \mathbf{F}_{nN_p\times n}\mathbf{X}_{n\times 1}\left(k\,|\,k\right) + \mathbf{\Phi}_{nN_p\times mN_c}\mathbf{U}_{mN_c\times 1} \tag{15}$$

where $\mathbf{X}\in R^{nN_p}$, $\mathbf{F}\in R^{nN_p\times n}$, $\mathbf{\Phi}\in R^{nN_p\times mN_c}$, $\mathbf{U}\in R^{mN_c}$ and $\mathbf{F},\mathbf{\Phi}$ are defined as

$$\mathbf{F}=\begin{bmatrix}\mathbf{A}\\ \mathbf{A}^2\\ \vdots\\ \mathbf{A}^{N_p-1}\\ \mathbf{A}^{N_p}\end{bmatrix}_{nN_p\times n}$$

$$\mathbf{\Phi}=\begin{bmatrix}\mathbf{B} & \bar{0} & \cdots & \bar{0}\\ \mathbf{AB} & \mathbf{B} & \cdots & \vdots\\ \vdots & \vdots & \ddots & \bar{0}\\ \mathbf{A}^{N_p-1}.\mathbf{B} & \mathbf{A}^{N_p-2}.\mathbf{B} & \cdots & \mathbf{A}^{N_p-N_c}.\mathbf{B}\end{bmatrix}_{nN_p\times mN_c} \tag{16}$$

The cost function is considered as

10.22060/EEJ.2025.23822.5635

$$J=\frac{1}{2}\mathbf{U}^T\mathbf{H}\mathbf{U}+(\mathbf{X}(k\,|\,k)-\mathbf{X}_d(k\,|\,k))\mathbf{M}\mathbf{U}$$
$$+\frac{1}{2}(\mathbf{X}(k\,|\,k)-\mathbf{X}_d(k\,|\,k))^T\mathbf{N}(\mathbf{X}(k\,|\,k)-\mathbf{X}_d(k\,|\,k)) \tag{17}$$

where $\mathbf{H}=2\left(\mathbf{\Phi}^T\mathbf{Q}\mathbf{\Phi}+\mathbf{R}\right),\mathbf{M}=2\mathbf{F}^T\mathbf{Q}\mathbf{F},\mathbf{N}=2\mathbf{F}^T\mathbf{Q}\mathbf{\Phi}$.

The control process aims to minimize the cost function, with the corresponding input signal being articulated as follows:

$$\mathbf{U}=-\mathbf{K}_{MPC}(\mathbf{X}(k\,|\,k)-\mathbf{X}_d(k\,|\,k)) \tag{18}$$

where $\mathbf{K}_{MPC}=\begin{bmatrix}\mathbf{I} & \bar{0} & \dots & \bar{0}\end{bmatrix}\mathbf{H}^{-1}\mathbf{M}^T$ and $\mathbf{X}_d(k\,|\,k)$ is a vector that guides the quadcopters to be in the right formation, following some rules.

In Fig 2, you can see the block diagram that shows how the quadcopter is controlled. The diagram highlights the importance of the control mechanism, MPC, which is crucial to making the quadcopter fly smoothly. Each part of the diagram is like a key player in controlling the quadcopter's flight. The use of MPC in this system is significant because it helps the quadcopter predict and adjust to changes in its surroundings.

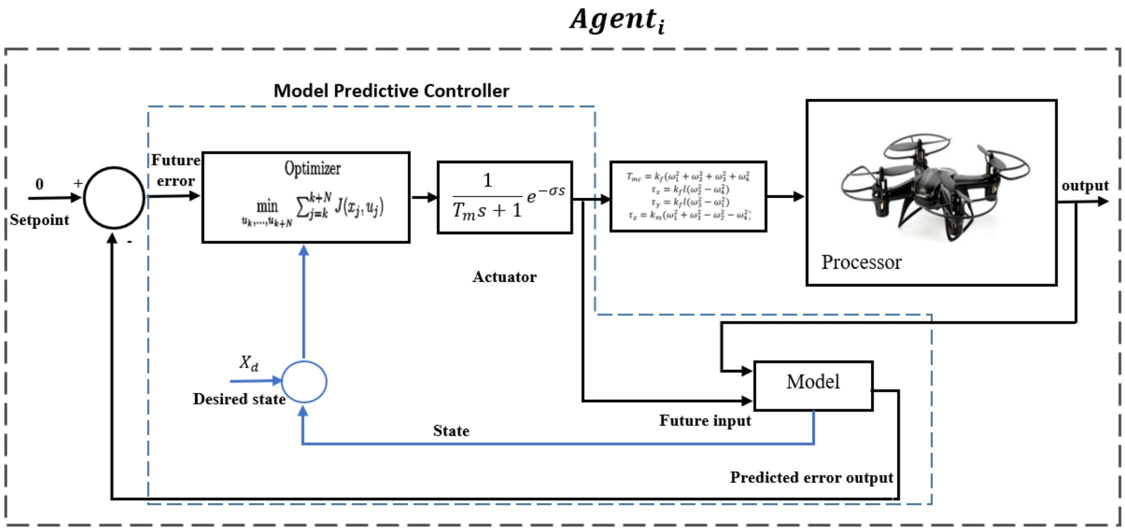

$Agent_i$
Model Predictive Controller
0
+
Setpoint
-
Future error
Optimizer
$\min_{u_k,\dots,u_{k+N}} \sum_{j=k}^{k+N} J(x_j,u_j)$
$\frac{1}{T_m s+1}e^{-\sigma s}$
Actuator
$T_{mc}=k_f(\omega_1^2+\omega_2^2+\omega_3^2+\omega_4^2$
$\tau_x=k_f l(\omega_2^2-\omega_4^2)$
$\tau_y=k_f l(\omega_3^2-\omega_1^2)$
$\tau_z=k_m(\omega_1^2+\omega_3^2-\omega_2^2-\omega_4^2$
Processor
output
X_d
Desired state
Model
State
Future input
Predicted error output

Fig 2. Block Diagram Quadcopter Control with MPC

The MPC controller is designed for all four quadcopters. We now develop formation rules in a multi-agent setup to plan the quadcopters' flight paths for their cooperative control. While the error-based MPC controller demonstrates strong robustness in tracking and uncertainty handling, it also comes with several practical limitations. First, the performance of MPC is highly sensitive to cost function tuning. Improper weight selection can lead to poor tracking, aggressive control signals, or instability. Second, the controller requires solving an optimization problem at every time step, which can be computationally demanding—especially as the number of agents, system states, or constraints increases. This makes MPC dependent on high-performance onboard processors, and in resource-limited systems, real-time implementation may be challenging or even infeasible. Third, MPC's performance can degrade under fast-changing environments or high-frequency disturbances if the prediction horizon or model accuracy is insufficient. Solver convergence time and numerical stability are also critical, particularly in embedded implementations. These limitations highlight the trade-offs between robustness and computational complexity, and they should be carefully considered when deploying MPC in real-world multi-agent UAV systems.

4. Formation and cooperative control

Multi-agent formation algorithm guides a group of autonomous agents, such as quadcopters, to achieve a desired formation shape for trajectory tracking. In this work, we consider a group of N quadcopters modeled as double integrators in 3D space:

$$\ddot{x}_{p_i} = u_{x_i},\ \ddot{y}_{p_i} = u_{y_i},\ \ddot{z}_{p_i} = u_{z_i} \tag{19}$$

The objective for the group of quadcopters is to achieve a desired formation shape defined by a set of desired relative positions [24]:

$$d_{ij} = \mathbf{p}_i - \mathbf{p}_j \tag{20}$$

where $\mathbf{p}_i = \left(x_{pi}, y_{pi}, z_{pi}\right)$ and d_{ij} is the desired distance between quadcopters i and j. The formation shape can be a square, triangle, line, etc., based on the relative positions.

The formation control law is designed based on graph theory, where each quadcopter is a node, and the desired distances are edges connecting the nodes. The control input for each quadcopter is calculated based on the desired distances to its neighboring quadcopters [25]:

$$u_{x_i} = \begin{cases} \sum\limits_{j\in N_i} a_{ij}(x_{pj} - x_{pi} - d_{x_{ij}}) + \gamma(x_{pd} - x_{pi}), & \text{if } i \text{ is leader} \\ \sum\limits_{j\in N_i} a_{ij}(x_{pj} - x_{pi} - d_{x_{ij}}), & \text{otherwise} \end{cases}$$

$$u_{y_i} = \begin{cases} \sum\limits_{j\in N_i} a_{ij}(y_{pj} - y_{pi} - d_{z_{ij}}) + \gamma(y_{pd} - y_{pi}), & \text{if } i \text{ is leader} \\ \sum\limits_{j\in N_i} a_{ij}(y_{pj} - y_{pi} - d_{y_{ij}}), & \text{otherwise} \end{cases} \tag{21}$$

$$u_{z_i} = \begin{cases} \sum\limits_{j\in N_i} a_{ij}(z_{pj} - z_{pi} - d_{z_{ij}}), & \text{if } i \text{ is leader} \\ \sum\limits_{j\in N_i} a_{ij}(z_{pj} - z_{pi}), & \text{otherwise} \end{cases}$$

where N_i is the set of neighboring quadcopters for quadcopter i and a_{ij} are control gains, and the desired path is defined, for instance $x_{pd} = rcos(\omega_r t)$ and $y_{pd} = rsin(\omega_r t)$.

After deriving the guiding law from the multi-agent formation principles, the desired state vector $\mathbf{X}_d$ is defined as follows:

$$\mathbf{X}_d = [0,0,0,0,0,0,x_p,0,y_p,0,z_p,0]$$

The control law minimizes the distance errors between neighboring quadcopters, driving the agents to the desired formation shape.

To demonstrate obstacle avoidance capabilities, an obstacle is introduced into the trajectory of the quadcopter formation. A repulsive potential field method enables the agents to maneuver around the obstacle and regain their original formation and trajectories. The control inputs in the equation (21) are redefined to incorporate the repulsive potential terms $u_{obsx,y,z}$ [26]:

$$u_{x_i} = \begin{cases} \sum\limits_{j\in N_i} a_{ij}(x_{pj} - x_{pi} - d_{x_{ij}}) + \gamma\left(x_{pd} - x_{pi}\right) + u_{obsx}, \text{ if } i \text{ is leader} \\ \sum\limits_{j\in N_i} a_{ij}(x_{pj} - x_{pi} - d_{x_{ij}}), \text{ otherwise} \end{cases}$$

$$u_{y_i} = \begin{cases} \sum\limits_{j\in N_i} a_{ij}(y_{pj} - y_{pi} - d_{z_{ij}}) + \gamma\left(y_{pd} - y_{pi}\right) + u_{obsy}, \text{ if } i \text{ is leader} \\ \sum\limits_{j\in N_i} a_{ij}(y_{pj} - y_{pi} - d_{y_{ij}}), \text{ otherwise} \end{cases} \tag{22}$$

$$u_{z_i} = \begin{cases} \sum\limits_{j\in N_i} a_{ij}\left(z_{pj} - z_{pi} - d_{z_{ij}}\right) + u_{obsz}, \text{ if } i \text{ is leader} \\ \sum\limits_{j\in N_i} a_{ij}(z_{pj} - z_{pi}), \text{ otherwise} \end{cases}$$

where $u_{obsx,y,z}$ represent the repulsive potential field forces defined as:

$$u_{obs_x} = \begin{cases} \frac{k}{r_1^2}(x_1 - x_o)\left(\frac{1}{r_1} - \frac{1}{r_a}\right), \text{ if } r_1 < r_a \\ 0, \text{otherwise} \end{cases}$$

$$u_{obs_y} = \begin{cases} \frac{k}{r_1^2}(y_1 - y_o)\left(\frac{1}{r_1} - \frac{1}{r_a}\right), \text{ if } r_1 < r_a \\ 0, \text{otherwise} \end{cases} \tag{23}$$

$$u_{obs_z} = \begin{cases} \frac{k}{r_1^2}(z_1 - z_o)\left(\frac{1}{r_1} - \frac{1}{r_a}\right), \text{ if } r_1 < r_a \\ 0, \text{otherwise} \end{cases}$$

where r_a is the activation radius, $[x_o, y_o, z_o]$ is the obstacle position and r_1 is the distance between the agent and obstacle defined as:

$$r_1 = \sqrt{(x_1 - x_o)^2 + (y_1 - y_o)^2 + (z_1 - z_o)^2} - r_d \tag{24}$$

where r_d is the obstacle diameter.

5. Simulation results

Fig 3 illustrates the topology of the quadcopter formation under consideration. Each quadcopter must have its internal controller to model these quadcopters as nodes in a graph theoretic framework. These controllers allow the quadcopters to follow prescribed formation laws with minimal error. The controller design utilizes the quadcopter and multi-agent system parameters provided in Table 1. A key parameter from Table 1, which enhances the model's fidelity, is the actuator delay. For all simulations, this was set to a realistic value of 10 Ms. An offline sensitivity analysis confirmed that our proposed MPC controller remains robustly stable for delays up to 50 Ms. Exceeding this value leads to a gradual degradation in tracking performance, thus establishing a clear stability margin for the control system.

Although we considered four agents to form a structured quadrilateral formation but it can be extended to any number of agents, the key factor is the communication and coordination framework among the agents, which can be adapted to different formation patterns regardless of the number of quadcopters.

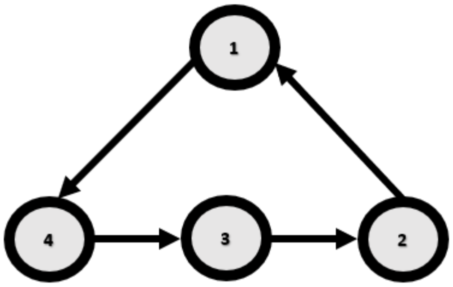


Fig 3. Interconnection Topology

Furthermore, the network's connectivity shares the formation's reference signal. Even if only one agent directly receives the leader's command, the communication graph represented by its Laplacian matrix,

10.22060/EEJ.2025.23822.5635

$$\mathbf{L}=\begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 1 \end{bmatrix}$$

ensures that the command spreads to neighboring agents. In this matrix, the values capture how each agent influences its neighbors, so each node adjusts its state based on the differences with its adjacent nodes. This leads to consensus across the network, keeping the formation intact and allowing for coordination regardless of the number of agents.

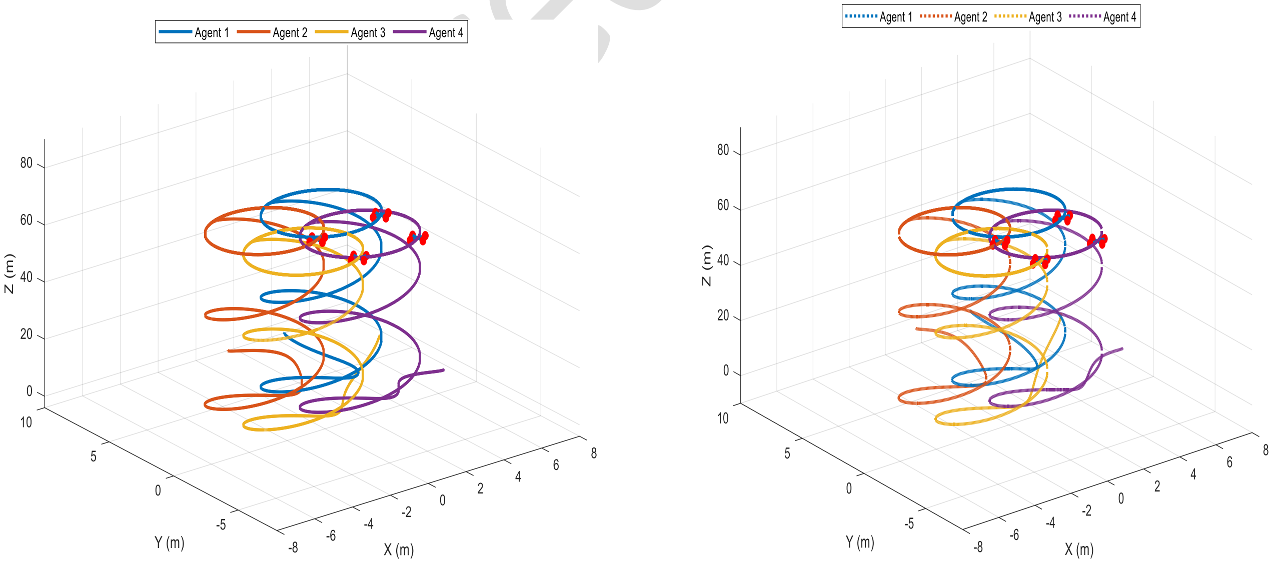
Table 1. Quadcopter model parameters [21]

Parameter	**value**
g	$9.81\left(\frac{\mathrm{m}}{\mathrm{s}^2}\right)$
m	$1.0230\,(\mathrm{kg})$
I_x	$0.0095\,(\mathrm{kg{\cdot}m^2})$
I_y	$0.0095\,(\mathrm{kg{\cdot}m^2})$
I_z	$0.0186\,(\mathrm{kg{\cdot}m^2})$
I_{xz}	$0\,(\mathrm{kg{\cdot}m^2})$
k_f	$1.4865\mathrm{e}-07$
k_m	$2.9250\mathrm{e}-09$
l	$0.2223\,(\mathrm{m})$
$\mathbf{Q}$	$10000*\mathrm{eye}(12)$
$\mathbf{R}$	$100*\mathrm{daig}(4)$
T_m	$200\,(\mathrm{ms})$
σ	$10\,(\mathrm{ms})$
N_p	100

N_c	30
a_{ij}	1
$d_{x_{12}}$	$3\,(\mathrm{m})$
$d_{x_{34}}$	$-3\,(\mathrm{m})$
$d_{y_{23}}$	$3\,(\mathrm{m})$
$d_{y_{41}}$	$-3\,(\mathrm{m})$
$\left[d_{x_{23}}, d_{x_{41}}, d_{y_{12}}, d_{x_{34}}\right]$	0
$d_{z_{12}}$	$\text{for} \begin{cases} 2, \ t < 150 \\ 0, \ t \geq 150 \end{cases}$
r	$3\,(\mathrm{m})$
ω_r	0.1
γ	1
$\mathbf{x}_{0agenq1}$	$[0, 0, 0, 0, 0, 0, 4 ,0, 9, 0, 0, 0]^{\mathrm{T}}$
$\mathbf{x}_{0agenq2}$	$[0, 0, 0 ,0 ,0 ,0, 1, 0 ,9, 0, 0, 0]^{\mathrm{T}}$
$\mathbf{x}_{0agenq3}$	$[0 ,0, 0, 0, 0, 0, 7 ,0 ,6 ,0, 0, 0]^{\mathrm{T}}$
$\mathbf{x}_{0agenq4}$	$[0, 0, 0, 0 ,0, 0, 7, 0, 1, 0, 0 ,0]^{\mathrm{T}}$
s	$0.0387\,(\mathrm{m}^2)$
C_d	0.1
$\mathbf{v}_W$	$[15 ,10 ,2]^{\mathrm{T}} \ \left(\frac{\mathrm{m}}{\mathrm{s}}\right)$
$[x_o, y_o, z_o]$	$[-1, -1 ,40]^{\mathrm{T}}$
r_d	2
r_a	10
k	100

5-1- Scenario 1

To evaluate the robustness of the LQR and MPC control algorithms with modeling uncertainty, the mass of each quadcopter is changed by 20% relative to the nominal design model. With this perturbation in the system dynamics, the 3D trajectory tracking performance of the two controllers is compared. Fig 4 shows the reference 3D trajectory tracked by the quadcopters controlled by the LQR and MPC algorithms. As can be seen, both controllers are able to track the desired path despite the uncertainty in the quadcopter models. The first agent, which acts as the leader, is visually distinguished by the blue trajectory in Fig 4 . Its corresponding time-dependent positions in the x, y, and z directions are shown in Fig 5 provides a comparison of the position tracking of the two algorithms, demonstrating that both achieved almost similar trajectory tracking in x, y, and z directions. Finally, Fig 6 compares the control signals generated by the LQR and MPC controllers. The results show that both control schemes could maintain stability and achieve adequate tracking performance despite uncertainties in the quadcopter mass parameters. This highlights the inherent robustness of the LQR and MPC approaches.



(a)

(b)

Fig 4. Comparison of Quadcopters Flight Paths with 20% Mass Uncertainty: a - LQR Controller, b - MPC Controller

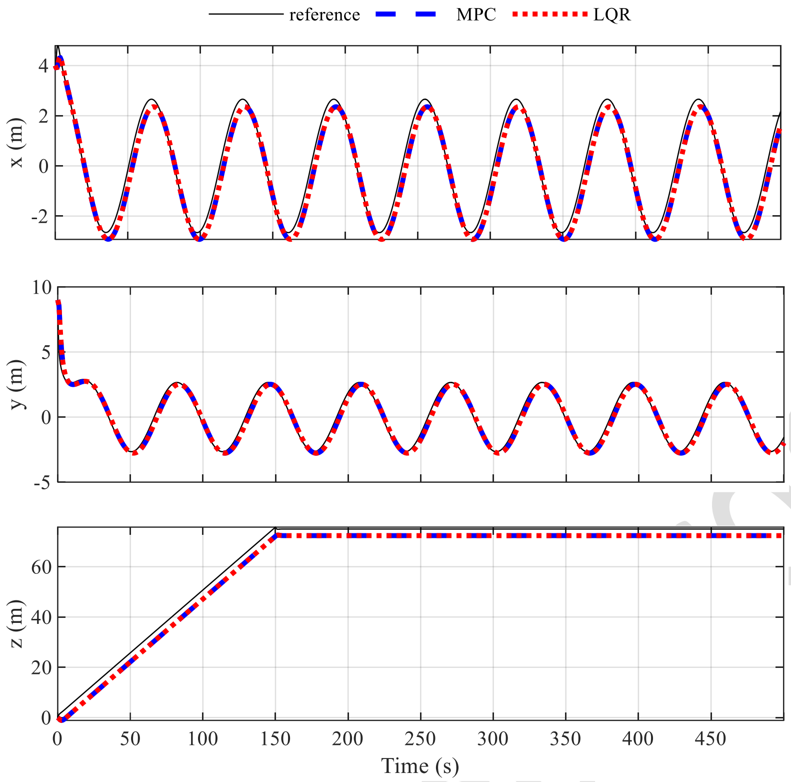


Fig 5. Time-Dependent Positions of Agent 1 with 20% Mass Uncertainty: A Comparison between LQR and MPC Controllers

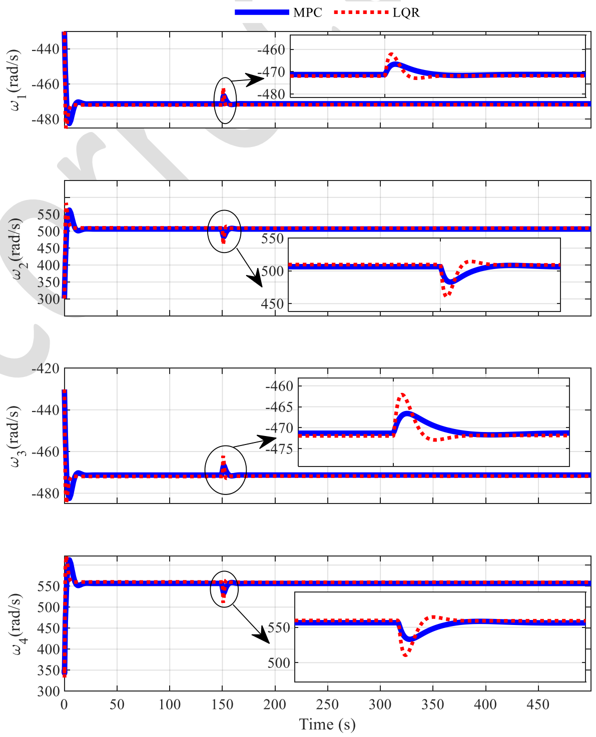


Fig 6. Control Signal Comparison for ω_1 to ω_4 Actuators with 20% Mass Uncertainty: LQR vs MPC Controllers over Time

10.22060/EEJ.2025.23822.5635

As shown by the trajectory tracking results, the LQR and MPC controllers exhibited robustness to modeling uncertainties in the quadcopter mass parameters. Despite a 20% change in the quadcopter mass relative to the nominal design model, the two controllers could adequately track the reference 3D path with minimal deviation from the desired trajectory. These results validate the ability of the LQR and MPC algorithms to perform well even when the real-world system dynamics do not perfectly match the models used for control design.

In this scenario, we assume that in the first factor one motor suffers a 20% breakdown, and Fig 7 and Fig 8 demonstrate that even under this condition the MPC controller maintains formation integrity and stability exceptionally well, whereas the LQR controller's performance degrades evidently.

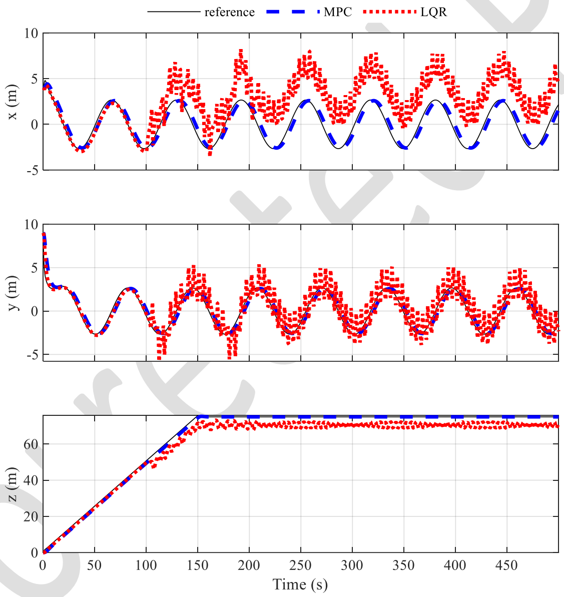


Fig 7. Positions of Agent 1 under 20% Motor Degradation and 20% Mass Uncertainty: A Comparison between LQR and MPC Controllers

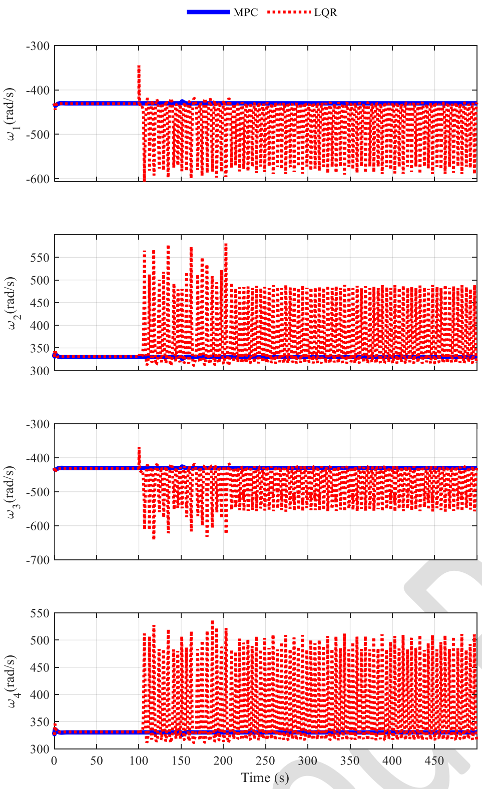
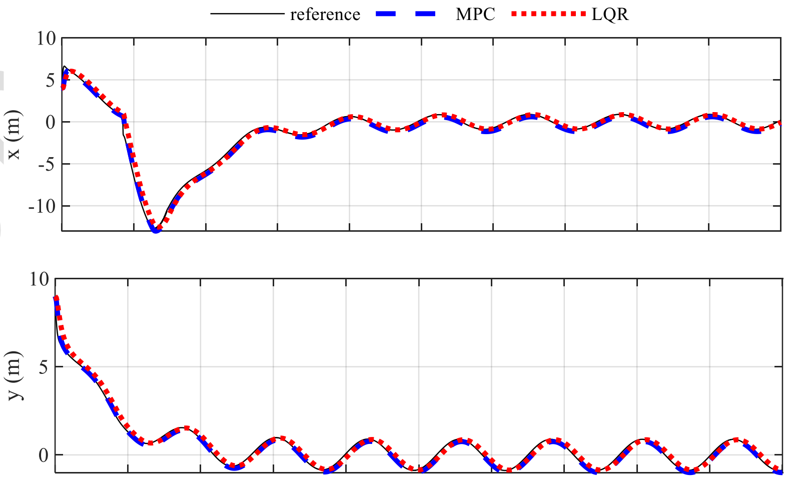


Fig 8. Speeds of Agent 1 under 20% Motor Degradation and 20% Mass Uncertainty: A Comparison between LQR and MPC Controllers

5-2- Scenario 2

The quadcopter states for this case are presented in Fig 9 , showing the formation regaining the original trajectory after bypassing the obstacle.



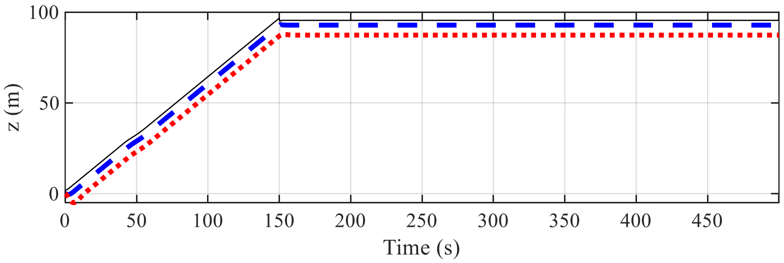


Fig 9. Time-Dependent Positions of Agent 1 with 20% Mass Uncertainty and Obstacle: A Comparison between LQR and MPC Controllers

The control signals generated by the LQR and MPC controllers to achieve obstacle avoidance are shown in Fig 10.

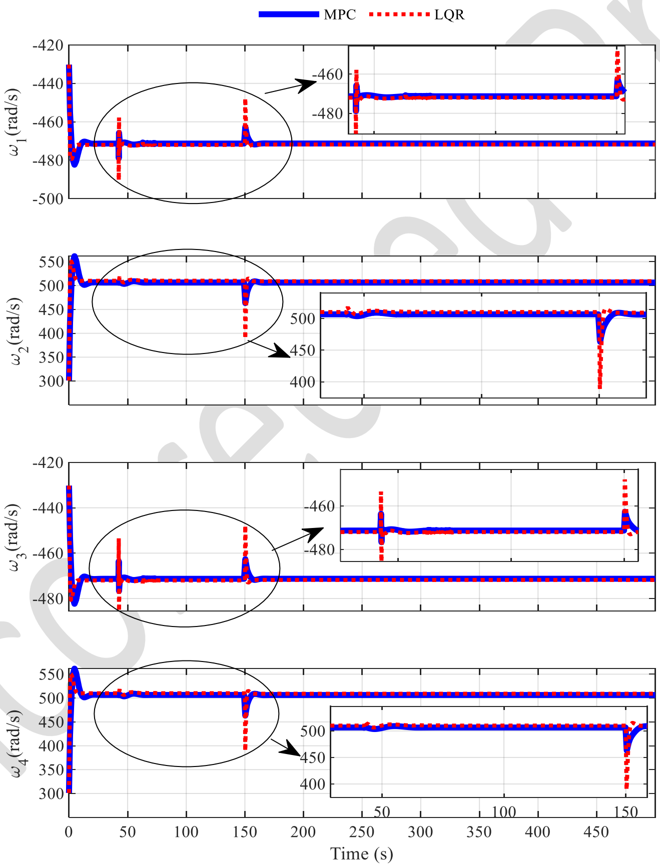


Fig 10. Control Signal Comparison for ω_1 to ω_4 Actuators with 20% Mass Uncertainty and Obstacle: LQR vs MPC Controllers over Time

Based on the quadcopter state responses shown in Fig 9, the MPC controller exhibited better transient performance for obstacle avoidance compared to the LQR controller, as evidenced by the smaller deviations in the states when bypassing the obstacle.

5-3- Scenario 3

In this scenario, we apply a sinusoidal disturbance exactly when it bypasses the obstacle, and the results can be seen in Fig 11. According to the results of this scenario, it is clear that the MPC controller can behave better than the LQR controller. From Fig 11, we observe that the height output controlled by the MPC is smoother and maintains stability better than the LQR controller. Fig 12 further supports this conclusion, showcasing the control signal output for the first agent. The MPC controller achieves a lower peak and uses less energy than the LQR controller, highlighting its efficiency and effectiveness.

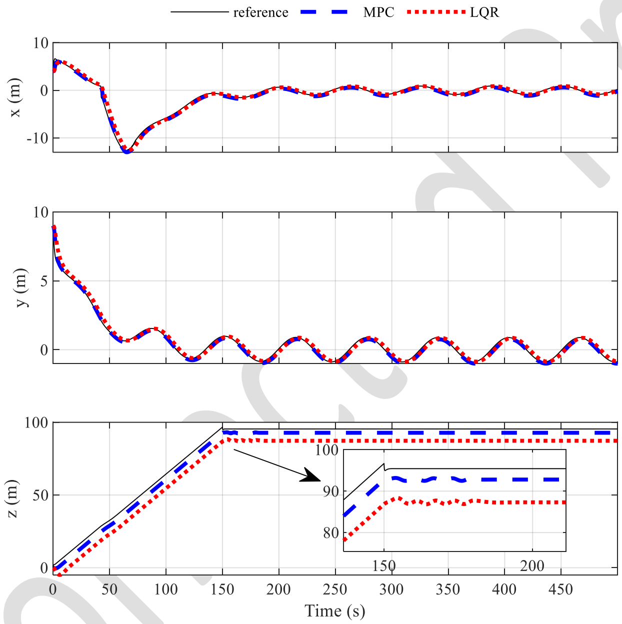


Fig 11. Time-Dependent Positions of Agent 1 with 20% Mass Uncertainty, Obstacle and Disturbance: A Comparison between LQR and MPC Controllers

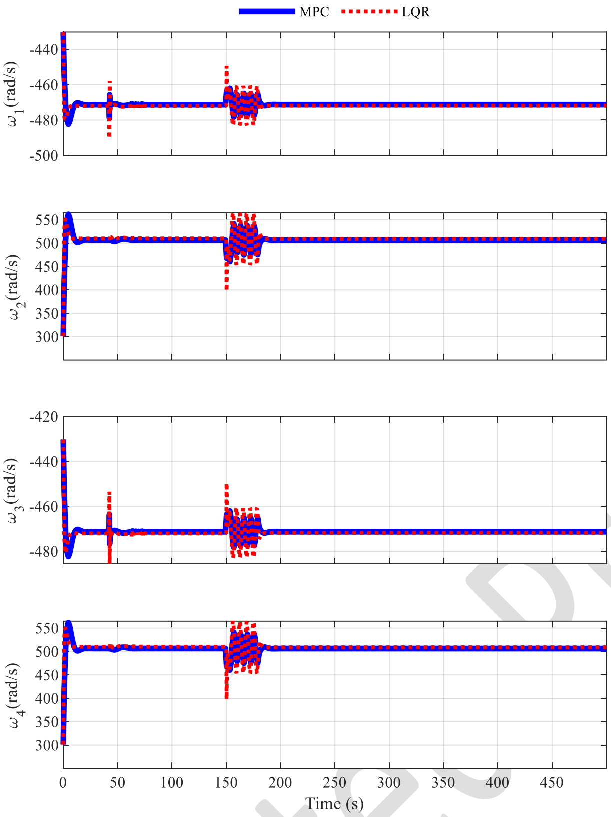


Fig 12. Control Signal Comparison for ω_1 to ω_4 Actuators with 20% Mass Uncertainty, Obstacle and Disturbance: LQR vs MPC Controllers

For a better comparison in this case, the Euler angles of the agents considering obstacle and uncertainty are also shown in Fig 13 . These figures show that the angles are controlled much better in the MPC controller, providing a more stable and precise response than the LQR controller. This indicates that the MPC controller is more robust in handling uncertainties and disturbances in dynamic environments.

10.22060/EEJ.2025.23822.5635

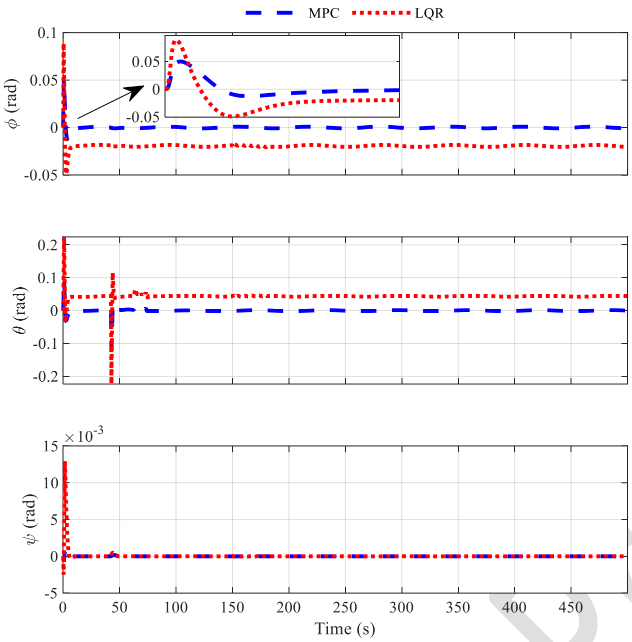


Fig 13. Attitudes of Agent 1 with 20% Mass Uncertainty, Obstacle and Disturbance: A Comparison between LQR and MPC

The presented results demonstrate the comparative performance of MPC and LQR controllers under the specified scenario involving sinusoidal disturbances and obstacles .Overall, the analysis confirms that the MPC controller performs better than the LQR controller in scenarios involving sinusoidal disturbances and obstacles. The MPC controller's ability to maintain stability, reduce energy consumption, and provide precise control makes it a better choice for applications requiring robust performance under uncertainty.

According to equation (20), the formation error is defined as

$$e_{ij} = | d_{ij} | - r_{ij}$$

where r_{ij} is the relative distance between agent i and agent j

$$r_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2}$$

To provide a clear and comprehensive evaluation, we also analyzed formation errors between neighboring agents in scenario three (Fig 14–Fig 17), offering a direct comparison of MPC, LQR, and PID performance. These plots reveal two periods of notable error increases: between 40 s and 85 s, when agents navigated around an obstacle—MPC showed superior transient performance and a faster return to optimal formation—and between 150 s and 190 s, during braking maneuvers that demanded precise formation integrity, where MPC again achieved the lowest errors, underscoring its robustness under dynamically varying conditions.

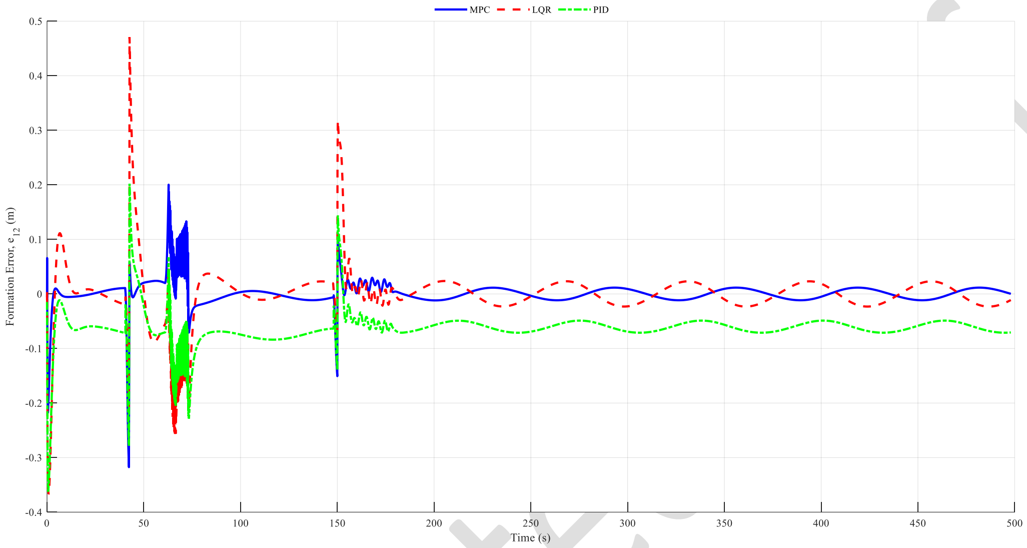


Fig 14. Formation Error between Agents 1 and 2 with 20% Mass Uncertainty, Obstacle and Disturbance: A Comparison between LQR, MPC and PID Controllers

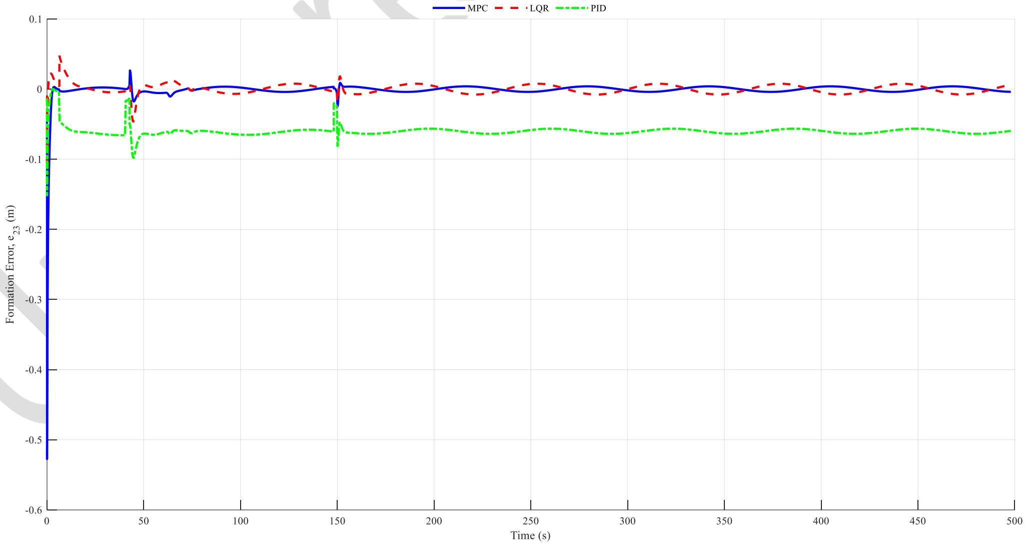


Fig 15. Formation Error between Agents 2 and 3 with 20% Mass Uncertainty, Obstacle and Disturbance: A Comparison between LQR, MPC and PID Controllers

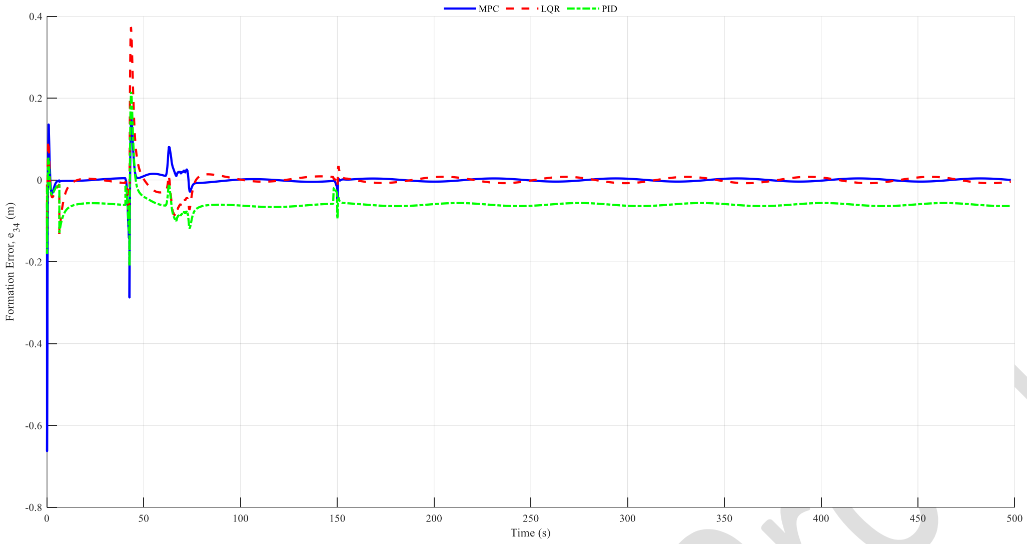


Fig 16. Formation Error between Agents 3 and 4 with 20% Mass Uncertainty, Obstacle and Disturbance: A Comparison between LQR, MPC and PID Controllers

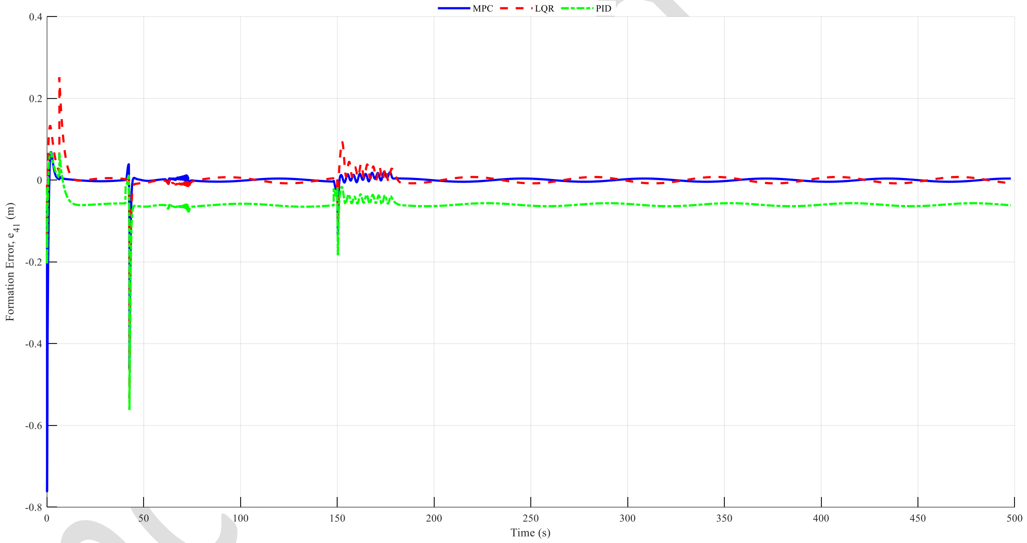


Fig 17. Formation Error between Agents 4 and 1 with 20% Mass Uncertainty, Obstacle and Disturbance: A Comparison between LQR, MPC and PID Controllers

6. Conclusion

This study presents a comprehensive framework for controlling and coordinating quadcopter systems, addressing nonlinear dynamics, actuator modeling, and environmental uncertainties with exceptional precision and adaptability. The nonlinear equations were effectively linearized through meticulous integration of wind drag effects, actuator dynamics, and delays, enabling the design of advanced controllers tailored to real-world challenges. The simulation results and comparative analyses demonstrate the superior performance of the error-based MPC controller over LQR, particularly in achieving precise trajectory tracking, robust obstacle avoidance, and energy-efficient control signals. The diagrams and results affirm that the MPC controller maintained remarkable stability and smooth responses under scenarios involving uncertainty, disturbances, and complex trajectories, outperforming LQR in all critical metrics. Extending this framework to the multi-agent system of four quadcopters, the proposed approach successfully managed dynamic formations, maintaining square geometries while navigating intricate helical paths and avoiding obstacles. These results validate the proposed system's robustness, flexibility, and effectiveness, showcasing its potential for deployment in real-world applications such as rescue operations, autonomous transport, and collaborative missions. This research offers a scalable, reliable, and highly adaptable control architecture, setting the stage for further advancements in multi-agent UAV coordination. We plan to improve the system’s robustness against attacks and communication failures for future work. This includes developing secure consensus and resilient control methods to detect and ensure stable formation control even when some agents are compromised.

References

[1] A. Katiar, R. Rashdi, Z. Ali, U. Baig, Control and stability analysis of quadcopter, in: 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), IEEE, 2018, pp. 1-6.

[2] O. Harkare, R. Maan, Design and Control of a Quadcopter, Int. J. Eng. Tech. Res, 10(5) (2021).

[3] V.K. Tripathi, L. Behera, N. Verma, Design of sliding mode and backstepping controllers for a quadcopter, in: 2015 39th national systems conference (NSC), IEEE, 2015, pp. 1-6.

[4] X. Luo, N. Han, X. Guan, Leader-following consensus protocols for formation control of multi-agent network, Journal of Systems Engineering and Electronics, 22(6) (2011) 991-997.

[5] Z. Lin, L. Wang, Z. Han, M. Fu, Distributed formation control of multi-agent systems using complex Laplacian, IEEE Transactions on Automatic Control, 59(7) (2014) 1765-1777.

10.22060/EEJ.2025.23822.5635

[6] J. Bai, G. Wen, A. Rahmani, Y. Yu, Formation tracking of fractional-order multi-agent systems based on error predictor, in: The 27th Chinese Control and Decision Conference (2015 CCDC), IEEE, 2015, pp. 279-284.

[7] Y.Q. Chen, Z. Wang, Formation control: a review and a new consideration, in: 2005 IEEE/RSJ International conference on intelligent robots and systems, IEEE, 2005, pp. 3181-3186.

[8] H.G. Tanner, On the controllability of nearest neighbor interconnections, in: 2004 43rd IEEE conference on decision and control (CDC)(IEEE Cat. No. 04CH37601), IEEE, 2004, pp. 2467-2472.

[9] P. Ogren, M. Egerstedt, X. Hu, A control Lyapunov function approach to multi-agent coordination, in: Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No. 01CH37228), IEEE, 2001, pp. 1150-1155.

[10] M.H. Yamchi, R.M. Esfanjani, Formation control of networked mobile robots with guaranteed obstacle and collision avoidance, Robotica, 35(6) (2017) 1365-1377.

[11] L. Sabattini, C. Secchi, C. Fantuzzi, Potential based control strategy for arbitrary shape formations of mobile robots, in: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2009, pp. 3762-3767.

[12] K.H. Kowdiki, R.K. Barai, S. Bhattacharya, Leader-follower formation control using artificial potential functions: A kinematic approach, in: IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM-2012), IEEE, 2012, pp. 500-505.

[13] N. Nfaileh, K. Alipour, B. Tarvirdizadeh, A. Hadi, Formation control of multiple wheeled mobile robots based on model predictive control, Robotica, 40(9) (2022) 3178-3213.

[14] S. Vargas, H.M. Becerra, J.-B. Hayet, MPC-based distributed formation control of multiple quadcopters with obstacle avoidance and connectivity maintenance, Control Engineering Practice, 121 (2022) 105054.

[15] L. Hu, K. Tian, C. Bai, Quadcopter Unmanned Aerial Vehicle Formation Trajectory Tracking based on Extended State Observer with Model Predictive Control, in: 2024 7th International Conference on Robotics, Control and Automation Engineering (RCAE), IEEE, 2024, pp. 240-246.

10.22060/EEJ.2025.23822.5635

[16] M. Tayefi, Z. Geng, Logarithmic control, trajectory tracking, and formation for nonholonomic vehicles on Lie group SE (2), International Journal of Control, 92(2) (2019) 204-224.

[17] S.M. Zakipour Bahambari, S. Khankalantary, Formation Control of Quadcopters with Unknown Target Tracking and Obstacle avoidance Using PID and MPC Controller, International Journal of Industrial Electronics Control and Optimization, 8(2) (2025) 165-176.

[18] J. Diebel, Representing attitude: Euler angles, unit quaternions, and rotation vectors, Matrix, 58(15-16) (2006) 1-35.

[19] S. Lu, Modeling, control and design of a quadrotor platform for indoor environments, Arizona State University, 2018.

[20] H. Bouadi, M. Tadjine, Nonlinear observer design and sliding mode control of four rotors helicopter, International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering, 1(7) (2007) 354-359.

[21] Q. Quan, Introduction to multicopter design and control, Springer, 2017.

[22] D.E. Kirk, Optimal control theory: an introduction, Courier Corporation, 2004.

[23] L. Wang, Model Predictive Control System Design and Implementation Using MATLAB, in, Springer Verlag, 2009.

[24] K.-K. Oh, M.-C. Park, H.-S. Ahn, A survey of multi-agent formation control, Automatica, 53 (2015) 424-440.

[25] W. Ren, Y. Cao, Distributed coordination of multi-agent networks: emergent problems, models, and issues, Springer Science & Business Media, 2010.

[26] G. Hao, Q. Lv, Z. Huang, H. Zhao, W. Chen, Uav path planning based on improved artificial potential field method, Aerospace, 10(6) (2023) 562.