MAAREF UNIVERSITY RESEARCH JOURNAL RJMU

Research Journal of Maaref University of Applied Sciences (RJMU)

Vol 1, Issue 1, pages.39-52, August (2025) **ISSN:** 0000-0000

Research Paper

A Novel Swarm-UAV Formation Tracking Control Based on Networked-MPC and Dual Quaternion Algebra

Dalia Kass Hanna [1] **Bassem Fares** [1] **Abdulkader Joukhadar** [2]

1. Department of Control Engineering and Automation, Faculty of Electrical and Electronics Engineering, University of Aleppo, Syria.
2. Department of Mechatronics Engineering, Faculty of Electrical and Electronics Engineering, University of Aleppo, Syria.

**Corresponding Author: Abdulkader Joukhadar:* e-mail: ajoukhadar@alepuniv.edu.sy

Available online at: https://journal.mas.edu.sy/MASV01E01D08202507

Received: 14 March 2025 Revised: 09 May 2025 Accepted: 26 July 2025

***Abstract*—** Notions of natural systems inspired the robotics and communication communities to develop a new path of research named "The Robotics Swarm Networked Control" to cooperatively handle vast areas of tasks well beyond a single robot's abilities. The cornerstone research topic in such systems is the formation tracking control. This article investigates a newly developed formation tracking control for a swarm of crewless aerial vehicles (swarm-UAV). The proposed controller is based on a model predictive control algorithm (MPC). The primary motivation for considering such a solution is the ability to handle constrained nonlinear multi-input multi-output systems (MIMO), such as swarm-UAV. To enable swarm-UAV to accurately track time-varying shapes in 3D space, dual quaternion algebra was proposed to model the swarm-UAV system, which provides a compact formula to represent the UAV's position and orientation without singularities. The proposed approach adopts graph theory to simulate the effects of the communication links between the UAVs. Simulation experiments for three scenarios using the CasADi package integrated with the MATLAB environment were conducted and discussed to highlight the effectiveness of the proposed controller for real-time implementation. Finally, some suggestions for future improvements and research lines are listed.

***Keywords*—** Swarm-UAV; DQ-Net-MPC; Formation tracking control.

I. INTRODUCTION

Background and Motivation

In recent years, robot swarms in cooperative tasks have skyrocketed to increase task performance with minimal human involvement. The system consists of multiple robots that can interact with each other and the environment to accomplish everyday tasks cooperatively. Examples of these tasks include: urban area exploration (Gunn & Anderson, 2015); search and rescue operations (Chitikena et al, 2023); urban firefighting (Perez-Saura et al, 2023); huge object transportation (Hartmann, 2023), etc.

A speedup in these tasks can be achieved by deploying a swarm of robots performing tasks in parallel. Fault Tolerant is crucial for any swarm system, which means a single robot client failure in the swarm system should not necessarily lead to the failure of the overall task (Miller & Gandhi, 2021). Such systems are called super-additive, where simple robots can perform complex goals, meaning the whole system can do more than a single one (Liu et al, 2022). Swarms of robots are classified into several groups, e.g., homogeneous and heterogeneous; large-scale and small-scale; and cooperative and competitive systems modes (Ajwad, 2020). For example, a swarm of uncrewed aerial vehicles (swarm-UAV) is a homogenous, large-scale system, and all UAVs work cooperatively to achieve a common goal.

Particularly, swarm-UAVs have drawn much research attention, providing an efficient solution, especially in real-time applications, thanks to their small size, ease of use, rapid technological advancement, and fast deployable flight (Alsammak et al, 2022). The robustness of swarm-UAV is strictly related to accurate mathematical modelling, communication network topology, and control structure used to reach the desired swarm behavior (i.e., formation control) (Stolfi & Danoy, 2022).

The most significant challenge researchers encounter when studying swarm-UAV is accurately modeling the individual and collective behavior coordination to ensure that the system works effectively as an integrated system (Sial et al, 2022). Using a simple mathematical formula to describe each UAV's altitude and attitude is vital to accomplishing an efficient formation control system in real-world applications, considering the kinematic limitations and the singularities (Nekoo et al, 2022). The most up-to-date studies in the field of swarm of robots modeling have utilized the dual quaternion algebra (DQ) (Savino et al, 2019). It is an extension of the quaternion algebra which is used to study both the position and orientation of rigid bodies in 3D space (Adorno & Marinho, 2021). DQ has strong algebraic properties to model complex robotics systems as a swarm of robots, UAVs, mobile manipulators (Mas & Kitts, 2018), and spacecraft (Valverde & Tsiotras, 2018).

The second challenge is the communication system, which involves a stable connection between UAVs. Such a system often requires information exchange among the UAVs using a networked control structure (Zaouche, 2017). The structure of networked control is traditionally categorized as centralized, decentralized, and distributed control (Azari, 2019). Centralized control consists of a global central control unit that connects to each UAV.

On the other hand, in the decentralized control, each UAV is controlled by independent local control units without any information exchange between UAVs. Finally, in the distributed control, all the UAVs are directly communicating with their neighbors and

Exchanging local information using communication networks (Lunze, 2014). In the case of a large-scale swarm, distributed control has more efficiency, adaptability, robustness, and scalability compared with centralized control, which has drawbacks like single point failure and limited computing capabilities (Zhuang et al, 2022). Recent approaches highlighted by researchers to control a swarm of robots are based on graph control theory. It models the swarm as massless points, which communicate through direct and indirect links.
The third challenge is the swarm control system. Generally, to achieve a common goal of the swarm of robots, each robot controller is implemented in three different levels: a task-level control (Qin et al, 2022), devoted to the calculus of the swarm strategy; a formation-level control (Hirata-Acosta et al, 2021), that considers the kinematic robot model; and dynamic control (Joukhadar et al, 2019), which controls the robot's actuators.
Researchers have proposed many kinematics and dynamics nonlinear controllers, such as sliding mode control (SMC), backstepping control (Joukhadar et al, 2019). Formation control is well studied and applied using different approaches due to its essential role for many complex tasks like human cooperative load transportation (Elwin et al, 2023). Thus, implementing a new solution for the formation control problem is an active area of research for different scenarios.
Recently, one of the most promising approaches for swarm formation control is model predictive control (MPC) (Stomberg et al, 2023).
MPC is an online optimization framework for a constrained nonlinear MIMO system. It is a model-based approach that predicts the system's behavior and solves the optimization problem to find the optimum inputs, ensuring the system's output tracks the desired trajectory (Chang & Shiau, 2018).
This paper proposes a novel formation tracking control based on an optimization framework and DQ algebra.
The proposed controller can maintain fast time-varying shapes accurately using modified networked model predictive control with dual quaternion algebra (DQ-Net-MPC).

Literature Review

This section summarizes the exciting research-related background topics and the state of the art of currently available approaches for the swarm-UAV control system. The concept of formation control has been studied extensively in the literature with applications to the coordination of multiple mobile manipulator robots (Alonso-Mora et al, 2017), UAVs (Dubois & Suzuki, 2018), and mobile robots (Loria et al, 2016). The key issue in the coordination control is localization. The localization challenge relates to all the tasks and behaviors associated with knowing the location of each robot in the swarm, especially the leader. Localization has been studied extensively; some research has been done on localization using multiple robots, which is called cooperative localization (Khan & Moura, 2007). Statistical and probabilistic techniques are the most common tools used in all methods given by (Joukhadar et al, 2019), (Joukhadar et al, 2020).
Three main control frameworks have emerged to address the formation problem: behavior-based and potential fields, leader-followers, and graph theory. Behavior-based methods and artificial potential fields are often combined in the formation control of a swarm system.
In (Yang et al, 2020), an artificial potential field-based approach is used as a formation control for a swarm system with collision and obstacle avoidance. This method is also a fault-tolerant and scalable swarm formation control methodology (Yang & Ye, 2024).
In (Xuan-Mung & Hong, 2019), UAVs maintain some desired distance to their neighbors; one UAV is designated as a leader while the other UAVs are designated as followers. Authors in (Liu et al, 2020) address the problem of heterogeneous formation tracking control with a virtual leader.
Graph theory is an influential tool that researchers use to mathematically model the interconnections between robots, especially when using a switched communication network topology and a large-scale swarm system (Ajwad, 2020). Distributed control law uses direct and indirect graphs.
In (Ebel, 2021), students distributed control-based communication links modeled using a direct graph. (Savino et al, 2019) mentions that most of the proposed studies use graph theory to represent only rigid body positions without considering their attitude, which leads to unstable performance in real-world applications.
To solve this problem, Siciliano et al (2011) use dual quaternions to represent the motion of complex robotics systems as rigid bodies work cooperatively in 3D space. A dual quaternion is a compact mathematical formula that describes the position and orientation of the robot; it is less expensive than homogeneous transformation matrix HTM multiplications (Valverde & Tsiotras, 2018).
In (Giribet et al, 2021), the authors demonstrate that a UAV and ground vehicles can collaborate to perform reconnaissance and surveillance. Other references (Adorno & Marinho, 2021) proposed a decentralized formation control law based on dual quaternion representation of mobile manipulators. Figueredo (2016) suggests different advanced kinematics control approaches based on DQ and improves the state-of-the-art in robot manipulators' cooperative control.
In recent years, researchers have become increasingly interested in developing advanced control approaches using optimization frameworks to overcome uncertainty effects online, considering system constraints. One of the most used approaches is model predictive control (MPC) (Stomberg et al, 2023).
Authors (Ghaderi et al., 2024) proposed a quadrotor formation control based on MPC that was improved with static and dynamic obstacle avoidance using an artificial potential field.
(Aspragkathos et al, 2024) Utilizes MPC with a visual servoing system in surveillance and tracking scenarios to enable effective tracking of moving targets.
To overcome the problem of the MPC's high computational power requirement in real-time applications, authors in (Elhesasy et al, 2023) use a fast optimizer like CasADi. It is an open source tool for constrained nonlinear optimization problems (NLP), it provides a fast optimal solution to a variety of numerical routines, and can be easily integrated with C++, Python, and MATLAB (Andersson et al, 2019).
The intersection of MPC with networked systems as swarm-UAV resulting in a networked model predictive control Net-MPC (Li et al, 2022).

Contributions

Although a lot of research deals with the study of swarm formation control, implementing controllers found in the literature for real-world applications is challenging and not as straightforward as expected. Limitations, like inaccurate kinematics models, time delay, communication limits, and scalability, should be taken into consideration to accomplish effective formation control. To address the mentioned limitations, this research aims to develop a novel formation tracking control to enable a complex system like swarm-UAV to achieve a desired time-varying formation while tracking a predefined trajectory. The proposed controller takes into consideration the UAV's attitude and the system constraints.

Motivated by the discussion in section 1.2, a modified networked model predictive control algorithm using dual quaternion algebra (DQ-Net-MPC) is proposed and tested as a swarm-UAV formation tracking control.

Using Net-MPC to directly consider the kinematics constraints, communication limits, network topology, and time delay in the MPC optimization problem is possible.

A fast optimizer (e.g., CasADi) integrated with the MATLAB environment is used for real-time implementation of the proposed approach.

This paper considers that the communication network topology is modeled with a fixed undirected graph, and the communication quality is ideal between UAVs.

On the other hand, to achieve the optimal swarm-UAV with singularity-free behavior and minimum computation operations, dual quaternions algebra was proposed to model the swarm-UAV system kinematics (i.e., all the UAVs' position and attitude).

The remaining sections of this paper are organized as follows. The system modeling and description using dual quaternions algebra and graph theory are introduced in Section 2. Next, in section 3, a modified networked model predictive control approach is developed to synthesize a formation tracking control law for a swarm-UAV modeled using dual quaternion algebra. A simulation study is presented and tested using MATLAB in Section 4, and finally, the discussion and conclusions are summarized in Sections 5 and 6, respectively.

II. Problem Statement and System Description

A swarm-UAV formation tracking control system consists of four basic components: UAVs model, the communication network topology, the controller, and the swarm task (7). This paper addresses the following problem:

Consider a swarm-UAV system consisting of n UAVs labeled $q = \{q_1, q_2, \ldots, q_n\}$. Let $\underline{\mathbf{x}}_{\mathbf{i}}$ represents the DQ pose (position and orientation) of each UAV_i in the swarm. The UAVs are assumed to communicate using a fixed network topology modeled with an undirected graph $\mathcal{G}$. The objective is to implement a networked control structure by solving an optimization framework to ensure that the swarm of UAVs tracks a predefined trajectory while maintaining a specific formation shape with minimal deviation. The adopted steps to implement the proposed DQ-Net-MPC approach are explained in detail. First, the DQ-based kinematic modeling of the swarm-UAV, followed by the modified Net-MPC, are presented.

Mathematical Kinematics Modeling of a Swarm-UAV using DQ algebra

This section presents a swarm-UAV kinematic modeling using DQ algebra. DQ describes the UAV's motion by combining the position and the orientation in a single mathematical expression called unit dual quaternions. DQ has strong algebraic properties, facilitating motion representation and providing free singularity with simple mathematical representation (Siciliano et al., 2011).

DQ's structure simplifies control laws, making it a valuable mathematical tool for complex robotics system modeling as swarm-UAV and real-time control requirements (Asaamoning et al, 2011). The notation to represent the sets used in this paper is as follows:

$\mathbb{R}$ represents the real numbers set.

$\mathbb{H}$ represents the quaternions set.

$\mathbb{H}_p$ represents the pure quaternions set.

$\mathcal{H}$ represents the DQ set.

$\mathcal{H}_\text{p}$ represents the unit dual quaternions set.

Hamilton proposed quaternions in the 19th century to extend the algebra of complex numbers (Adorno & Marinho, 2021). The coefficients of a quaternion number $\mathbf{q} \in \mathbb{H}$ are given by:

$$\mathbf{q} \in \mathbb{H} \triangleq \{q_1 + \hat{\imath}q_2 + \hat{\jmath}q_3 + \hat{k}q_4 : q_1, q_2, q_3, q_4 \in \mathbb{R}\ and\ \hat{\imath}^2 = \hat{\jmath}^2 = \hat{k}^2 = \hat{\imath}\hat{\jmath}\hat{k} = -1\}, \tag{1}$$

The 4-tuple of independent real coefficients q_1, q_2, q_3, q_4 assigned to one real axis and three orthonormal imaginary axes $\hat{\imath},\ \hat{\jmath},\ \hat{k},$ These are called quaternions units.

DQ combines dual number theory and quaternions, offering a novel mathematical form with unique properties $\mathcal{H}$.

The DQ number $\underline{\mathbf{q}} \in \mathcal{H}$ consists of two quaternions $\mathbf{q}_{\mathcal{P}}, \mathbf{q}_{\mathcal{D}} \in \mathbb{H}$, (The primary part $\mathbf{q}_{\mathcal{P}} = \mathcal{P}(\underline{\mathbf{q}})$, the dual part $\mathbf{q}_{\mathcal{D}} = \mathcal{D}(\underline{\mathbf{q}})$) and an algebraic unit ε called "dual unit", satisfying the properties $\varepsilon^2 = 0$; $\varepsilon \neq 0$.

$$\underline{\mathbf{q}} \in \mathcal{H} \triangleq \{\mathbf{q}_{\mathcal{P}} + \varepsilon\mathbf{q}_{\mathcal{D}}\}, \tag{2}$$

Sometimes, it is necessary to multiply real matrices with dual quaternions. Thus, some appropriate mathematical operators are needed. The used operators in this paper are summarized below (Adorno & Marinho, 2021):

- The operator $vec_8 : \mathcal{H} \to \mathbb{R}^8$: This operator is used to map between the DQ set and the eight-dimensional vector space sets $\mathcal{H} \to \mathbb{R}^8$ and is defined as:

Given $\underline{\mathbf{q}} \in \mathcal{H}$ such that $\underline{\mathbf{q}} = q_1 + \hat{\imath}q_2 + \hat{\jmath}q_3 + \hat{k}q_4 + \varepsilon(q_5 + \hat{\imath}q_6 + \hat{\jmath}q_7 + \hat{k}q_8)$ Then:

$$vec_8(\underline{\mathbf{q}}) = [q_1 \quad q_2 \quad q_3 \quad q_4 \quad q_5 \quad q_6 \quad q_7 \quad q_8]^\text{T}, \tag{3}$$

- Hamilton operators $\overset{+}{\mathbf{H}}_8(\underline{\mathbf{q}}), \overset{-}{\mathbf{H}}_8(\underline{\mathbf{q}})$: Very useful tools when performing algebraic multiplications, these operators can be utilized to commute terms when dual quaternions numbers $\mathcal{H}$ are mapped into $\mathbb{R}^8$. The DQ Hamilton operators are defined as:

$$\overset{+}{\mathbf{H}}_8(\underline{\mathbf{q}}) = \begin{bmatrix} \overset{+}{\text{H}}_4(\mathbf{q}_{\mathcal{P}}) & \mathbf{0}_4 \\ \overset{+}{\text{H}}_4(\mathbf{q}_{\mathcal{D}}) & \overset{+}{\text{H}}_4(\mathbf{q}_{\mathcal{P}}) \end{bmatrix}, \quad \overset{-}{\mathbf{H}}_8(\underline{\mathbf{q}}) = \begin{bmatrix} \overset{-}{\text{H}}_4(\mathbf{q}_{\mathcal{P}}) & \mathbf{0}_4 \\ \overset{-}{\text{H}}_4(\mathbf{q}_{\mathcal{D}}) & \overset{-}{\text{H}}_4(\mathbf{q}_{\mathcal{P}}) \end{bmatrix}, \tag{4}$$

where the elements $\overset{+}{\text{H}}_4(\mathbf{q}_{\mathcal{P}}), \overset{-}{\text{H}}_4(\mathbf{q}_{\mathcal{P}})$ Are the quaternions Hamilton operators given by:

$$\overset{+}{\mathbf{H}}_4(\mathbf{q}) = \begin{bmatrix} q_1 & -q_2 & -q_3 & -q_4 \\ q_2 & q_1 & -q_4 & q_3 \\ q_3 & q_4 & q_1 & -q_2 \\ q_4 & -q_3 & q_2 & q_1 \end{bmatrix}, \qquad (5)$$
$$\overset{-}{\mathbf{H}}_4(\mathbf{q}) = \begin{bmatrix} q_1 & -q_2 & -q_3 & -q_4 \\ q_2 & q_1 & q_4 & -q_3 \\ q_3 & -q_4 & q_1 & q_2 \\ q_4 & q_3 & -q_2 & q_1 \end{bmatrix},$$

Using Hamilton operators, the mathematical expression for multiplying two DQ numbers $\underline{\mathbf{q}}_1\underline{\mathbf{q}}_2$ is written as follows:

$$vec_8(\underline{\mathbf{q_1 q_2}}) = \overset{+}{\mathbf{H}}_8(\underline{\mathbf{q}}_1)vec_8(\underline{\mathbf{q}}_2) = \overset{-}{\mathbf{H}}_8(\underline{\mathbf{q}}_2)vec_8(\underline{\mathbf{q}}_1), \qquad (6)$$

- DQ conjugate operator $\mathbf{C}_8$: This operator is used to express the conjugate of a given DQ number $\underline{\mathbf{q}}^*$ into $\mathbb{R}^8$ set as follows:

$$vec_8(\underline{\mathbf{q}}^*) = \mathbf{C}_8 vec_8(\underline{\mathbf{q}});\ \mathbf{C}_8 = diag(1,-1,-1,-1,1,-1,-1,-1), \qquad (7)$$

If the primary part of $\mathbf{q}_{\mathcal{P}}$ is unit quaternions (i.e. $\|\mathbf{q}_{\mathcal{P}}\| = \sqrt{\mathbf{q}_{\mathcal{P}}\mathbf{q}_{\mathcal{P}}^*} = 1$), and perpendicular to the dual part $\mathbf{q}_{\mathcal{P}}\mathbf{q}_{\mathcal{D}} = 0$, then $\underline{\mathbf{q}} \in \mathcal{H}_\mathrm{p}$ and is called unit dual quaternions.

To perform a kinematics study using DQ algebra, a compact DQ framework corresponding to a translation $\mathbf{p}$ followed by a rotation $\mathbf{r}$ is used to represent a rigid body motion in 3D-space in the form of a unit dual quaternion.

In this paper, the kinematics description of each $\mathrm{UAV_i}$ concerning a fixed coordinate frame, $\mathcal{O}$ is given by $\underline{\mathbf{x}}_\mathrm{i}^{\mathcal{O}} \in \mathcal{H}_\mathrm{p}$:

$$\underline{\mathbf{x}}_\mathrm{i}^{\mathcal{O}} = \mathbf{r}_\mathrm{i}^{\mathcal{O}} + \frac{1}{2}\varepsilon\mathbf{p}_\mathrm{i}^{\mathcal{O}}\mathbf{r}_\mathrm{i}^{\mathcal{O}};\ \mathbf{r}_\mathrm{i}^{\mathcal{O}} \in \mathrm{Spin}(3), \mathbf{p}_\mathrm{i}^{\mathcal{O}} \in \mathbb{H}_p, \qquad (8)$$

where $\mathbf{r}_\mathrm{i}^{\mathcal{O}} = \cos(\frac{\emptyset}{2}) + \mathbf{n}\sin(\frac{\emptyset}{2})$, is a unit quaternion that represents the rotation vector with angle $\emptyset$ around an arbitrary unit vector $\mathbf{n}$, and $\mathbf{p}_\mathrm{i}^{\mathcal{O}} = (\mathrm{p_x}\hat{\imath} + \mathrm{p_y}\hat{\jmath} + \mathrm{p_z}\hat{k})$ Represent the position vector as a pure quaternion (i.e., the fundamental part of $\mathbf{p}$ should be equal to zero). According to (Siciliano et al, 2011), the derivative of $\underline{\mathbf{x}}_\mathrm{i}^{\mathcal{O}}$ Can be expressed by:

$$\dot{\underline{\mathbf{x}}}_\mathrm{i}^{\mathcal{O}} = \frac{1}{2}\underline{\boldsymbol{\xi}}_\mathrm{i}^{\mathcal{O}}\underline{\mathbf{x}}_\mathrm{i}^{\mathcal{O}}, \qquad (9)$$

where $\underline{\boldsymbol{\xi}}_\mathrm{i}^{\mathcal{O}} \in \mathcal{H}_\mathrm{p}$ Is the twist $\mathrm{UAV_i}$ concerning the frame $\mathcal{O}$, and is defined as (Siciliano et al, 2011):

$$\underline{\boldsymbol{\xi}}_\mathrm{i}^{\mathcal{O}} = \boldsymbol{\omega}_\mathbf{i} + \varepsilon(\dot{\mathbf{p}}_\mathbf{i} + \mathbf{p}_\mathbf{i} \times \boldsymbol{\omega}_\mathbf{i}), \qquad (10)$$

where $\boldsymbol{\omega}_\mathbf{i} \in \mathbb{H}_p$ is the angular velocity and $\dot{\mathbf{p}}_\mathbf{i} \in \mathbb{H}_p$ Is the linear velocity. Finally, it is essential to mention that the discrete form of the kinematics equation (9) for the non-Euclidean properties of $\mathcal{H}_\mathrm{p}$ set is expressed by (Figueredo, 2016):

$$\underline{\mathbf{x}}_\mathrm{i}^{\mathrm{k}+1} = \exp(\frac{\mathrm{T_s}\underline{\boldsymbol{\xi}}_\mathrm{i}^\mathrm{k}}{2})\,\underline{\mathbf{x}}_\mathrm{i}^\mathrm{k}, \qquad (11)$$

where $\mathrm{T_s}$ is the sample time. The next section presents a detailed description of the networked control protocols and how to integrate these protocols with DQ-based modeling is presented.

Integrate Formation Control with DQ algebra.

Formation control is one of the most important research topics in Swarm-UAV. According to Iskender et al (2019), the swarm-UAV is a complex system consisting of multiple UAVs working cooperatively to complete a specific task by establishing practical networked control approaches depending on the information interaction protocol between UAVs. This protocol is used to determine and update the next state of the UAVs (Iskender et al, 2019).

In a distributed control, the swarm-UAV can be partially or fully connected with a fixed or switched communication topology between UAVs, which imposes constraints on the control system and should be considered (Seisa et al, 2022). Studiers use graph theory as a main tool to model the swarm communication topology mathematically.

The communication topology can be modeled by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$ the set of N nodes and $\mathcal{E} = \{(v_i, v_j)|v_i, v_j \in \mathcal{V}, i \neq j\} \subset \mathcal{V} \times \mathcal{V}$ the set of edges, where the edge $(v_i, v_j) \in \mathcal{E}$ represents the communication link from node i to j.

A node j is said to be a neighbor of node i if there is an edge. $(v_i, v_j) \in \mathcal{E}$. The neighbor set of the node i is denoted by $\mathcal{N}_i$, where $\mathcal{N}_\mathrm{i} := \{\upsilon_\mathrm{j}|\upsilon_\mathrm{i}, \upsilon_\mathrm{j} \in \mathcal{V}, (\upsilon_\mathrm{i}, \upsilon_\mathrm{j}) \in \mathcal{E}, \mathrm{i} \neq \mathrm{j}\} \subseteq \mathcal{V}$, which means that node i can receive information from any $j \in \mathcal{N}_i$.

The properties of graph $\mathcal{G}$ are reduced into three matrices, (i.e., adjacency matrix $\boldsymbol{\mathcal{A}}$, degree matrix $\boldsymbol{\mathcal{D}}$, and Laplacian matrix $\boldsymbol{\mathcal{L}}$). The adjacency matrix is used to describe the directional communication among the robots, which is defined as $\boldsymbol{\mathcal{A}} = [a_{ij}] \in \mathbb{R}^{M\times M}$ With each entry expressed as:

$$a_{ij} = \begin{cases} 1, & if\ (v_i, v_j) \in \mathcal{E} \\ 0, & otherwise \end{cases} \qquad (12)$$

The degree matrix $\boldsymbol{\mathcal{D}} = diag\,(d_{ii}) \in \mathbb{R}^{M\times M}$, with $\mathrm{d_{ii}}$ containing the information about the number of edges attached to each vertex. The Laplacian matrix of the graph $\mathcal{G}$ is defined by $\boldsymbol{\mathcal{L}} = \boldsymbol{\mathcal{D}} - \boldsymbol{\mathcal{A}} \in \mathbb{R}^{M\times M}$. The matrix $\boldsymbol{\mathcal{L}}$ plays an important role in the swarm control as shown next [18].

The graph theory distinguishes between undirected and directed graphs as follows: In the undirected graph, the edges connect the nodes without specifying the connection direction (i.e. $a_{ij} = a_{ji} = 1$), on the other hand, in the directed graph, each edge has a starting node and an ending node [18].

According to the graph theory, the most essential issue in cooperative networked control is the consensus protocol (or agreement protocol).

In the consensus protocol, the control input u_i^k for each $\mathrm{UAV_i}$ is represented by the rate change of the state $\dot{x}_i^k$ At time instance k, which is assumed to be governed by the sum of its relative states x_i^k for its neighbors x_j^k, as given by:

$$u_i^k = \dot{x}_i^k = -\sum_{j\in \mathcal{N}_i} a_{ij}\, x_{ij}^k = -\sum_{j\in \mathcal{N}_i} a_{ij}\left(x_j^k - x_i^k\right), \quad i = 1, \ldots, N, \; i \neq j, \tag{13}$$

Then, the discrete state space of $\mathrm{UAV_i}$ is given by:

$$x_i^{k+1} = x_i^k + T_s u_i^k, \quad i = 1, \ldots, N, \tag{14}$$

Where N is the number of UAVs, and $\mathcal{N}_i$ is the neighbor's set of the $\mathrm{UAV_i}$ in the swarm. Another form of equation (13) can be expressed using the Laplacian matrix $\boldsymbol{\mathcal{L}} \in \mathbb{R}^{n\times n}$ Of the swarm interaction graph $\mathcal{G}$ is given by:

$$\mathbf{u_k} = \mathbf{v_k} = -\boldsymbol{\mathcal{L}}\, \mathbf{x_k}, \tag{15}$$

where $\mathbf{u_k} \in \mathbb{R}^n$ Is the swarm control vector, $\mathbf{v_k} \in \mathbb{R}^n$ is the swarm velocities vector, and $\mathbf{x_k} \in \mathbb{R}^n$ Is the swarm's state vector. In some cases, UAVs must also track a reference trajectory generated by a real or virtual leader. This is referred to as leader tracking consensus.

A special case of tracking consensus is formation tracking. Figure 1 illustrates information tracking the swarm-UAV force to track the leader's trajectory while maintaining a specific shape by controlling the motion of all UAVs for a desired relative interpositions between UAVs (Iskender et al, 2019).

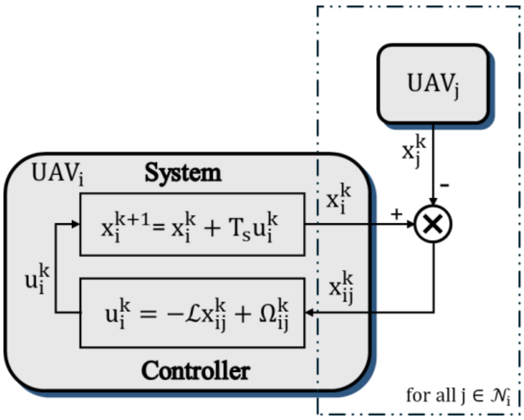


Figure (1). A block diagram of formation tracking control.

Define $\omega_k^{ij} \in \mathbb{R}^n$ which denote the desired relative position between $\mathrm{UAV_i}$ and $\mathrm{UAV_j}$, the formation tracking control protocol is given by:

$$u_i^k = -\sum_{j\in N_i} a_{ij}\left(x_{ij}^k - \omega_k^{ij}\right), \quad i = 1, \ldots, N, \tag{16}$$

Using the Laplacian matrix, equation (16) can be written as follows:

$$\mathbf{u_k} = -\boldsymbol{\mathcal{L}}\mathbf{x_k} + \boldsymbol{\Omega}_\mathbf{k}, \quad i = 1, \ldots, N, \tag{17}$$

where $\Omega_k = \left[\Omega_1^k, \Omega_2^k, \ldots, \Omega_n^k\right] \in \mathbb{R}^n$, and $\Omega_i^k = -\sum_{j\in N_i} \omega_k^{ij}$ for all $i = 1, \ldots, N$.

In case that the swarm-UAV kinematics was expressed using a unit dual quaternion (i.e. $\underline{\mathbf{x}}_i^{\mathcal{O}} \in \mathcal{H}_p$), the control protocol (17) cannot be applied directly because $\mathcal{H}_p$ is a non-Euclidean space as mentioned in [12].

To overcome this problem, researchers in [12] proposed to modify the system output by using the logarithm of $\underline{\mathbf{y}}_i = log(\underline{\mathbf{x}}_i) \in \mathcal{H}_p$. Thus, equation (13) is reformed in DQ space as follow:

$$vec_8(\underline{\mathbf{u}}_i) = vec_8(\dot{\underline{\mathbf{x}}}_i) = -\mathbf{Q_8}(\underline{\mathbf{x}}_i)\sum_{j\in \mathcal{N}_i} a_{ij} vec_6\left(\underline{\mathbf{y}}_i - \underline{\mathbf{y}}_j\right), \tag{18}$$

where $\mathbf{Q_8}(\underline{\mathbf{x}}_i) \in \mathbb{R}^{8\times 6}$ is the mapping matrix between the derivative of a unit dual quaternion and the derivative of its logarithm [13].

Section 3 addresses the formation control problem using DQ in the context of model predictive control (MPC). This novel control is proposed to find the optimal control law in DQ space with constraints.

III. DQ-Based Networked Model Predictive Control Approach

This section proposes a novel swarm-UAV formation tracking control approach based on DQ and networked model predictive control (DQ-Net-MPC). DQ-Net-MPC can be seen as the intersection of research fields of advanced control theory, mathematical optimization, and communications networks.
The block diagram for the proposed approach is shown in Figure 2. The kinematics description of the swarm-UAV system was expressed in DQ space, which facilitates the swarm motion representation with free singularity as mentioned in section 2.1. UAVs are dynamically decoupled. Thus, to maintain the swarm working cooperatively, the coupling is achieved in the control input using the properties of the communication graph G (i.e., the Laplacian matrix L). The control objective is to guarantee that the swarm-UAV tracks a predefined trajectory while maintaining a desired formation shape specified by the task controller.

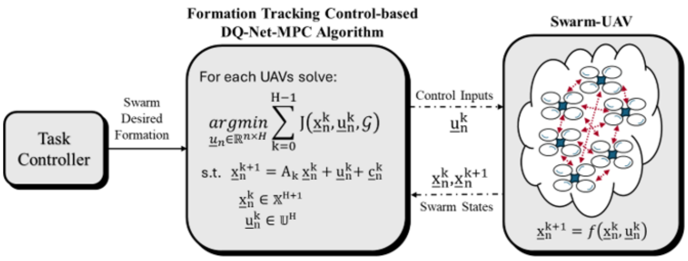


Figure (2): A block diagram of the proposed DQ-Net-MPC formation tracking control.

First, a brief introduction about the MPC algorithm is presented, then the problem formulation and the proposed solution are explained extensively.

- MPC algorithm

As a swarm-UAV, MPC is one of the most attractive control approaches for constrained multi-input, multi-outputs

(MIMO) systems. A feedback control approach uses model-based optimization to compute the control input in a repeated form [27]. MPC solves an optimization problem over a prediction horizon H with a given objective function J to minimize the tracking error and improve coordination (Borghi & Herty, 2023).

The physical limits are the maximum communication range, and the voltage or velocity limits are represented as constraints on the states and the control vectors, respectively [17].

Consider a linear discrete-time variant system given by:

$$\mathbf{x_{k+1}} = \mathbf{A_k}\mathbf{x}_k + \mathbf{B_k}\mathbf{u}_k, \tag{19}$$

where the sample instance $k \in [0,1,\dots,H-1]$, $\mathbf{x_k} \in \mathbb{R}^n$ Is the state vector at k, $\mathbf{u}_k \in \mathbb{R}^m$ Is the control input vector, $\mathbf{A_k} \in \mathbb{R}^{n\times n}$ Is the variant system matrix and $\mathbf{B_k} \in \mathbb{R}^{n\times m}$ Is the variant input matrix.

The cost function of the optimization problem is chosen as a quadratic sum of the states and control inputs as given by (20):

$$\mathrm{J}(\mathbf{x}_k,\mathbf{u}_k) = \mathbf{x}_H^T\mathbf{Q}_H\mathbf{x}_H + \sum_{k=0}^{H-1}\mathbf{x}_k^T\mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^T\mathbf{R}\mathbf{u}_k, \tag{20}$$

where $\mathbf{x_H} \in \mathbb{R}^n$ is the state vector at the end of the horizon H, $\mathbf{Q}_H, \mathbf{Q} \in \mathbb{R}^{n\times n}, \mathbf{R} \in \mathbb{R}^{m\times m}$ are positive symmetric weighting matrices. The state and control sequence over the predicted time horizon is defined as $\mathbf{X} = (\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_H), \mathbf{U} = (\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_{H-1})$.

The above cost function is subject to linear equality/inequality constraints on the states and the control inputs. The constraint sets can be expressed as (Borghi & Herty, 2023):

$$\begin{aligned}\mathbf{X} &\in \mathbb{X}^{H+1}\\ \mathbf{U} &\in \mathbb{U}^H\end{aligned} \tag{21}$$

These constraints can be used, for example, to incorporate the communication limitations (the maximum range of the communication network between UAVs) into the control framework as $[\boldsymbol{x_{min}}, \boldsymbol{x_{max}}]$ and the swarm-UAV velocities range as control input constraints as $[\boldsymbol{u_{max}}, \boldsymbol{u_{min}}]$.

A predicted sequence of optimal control actions $\boldsymbol{u_k^*} = (\boldsymbol{u_1^*}, \boldsymbol{u_2^*}, \dots, \boldsymbol{u_{H-1}^*})$ It is calculated by solving the following optimization problem for a specific prediction horizon H for each sample instance k:

$$\begin{aligned}\mathrm{u}_k^* &= \underset{\mathbf{u}_k \in \mathbb{R}^{n\times H}}{\operatorname{argmin}} \sum_{k=0}^{H-1}\mathrm{J}(\mathbf{x}_k,\mathbf{u}_k),\\ s.t.\quad \mathbf{x_{k+1}} &= \mathbf{A_k}\mathbf{x}_k + \mathbf{B_k}\mathbf{u}_k \qquad \textit{System Model}\\ \mathbf{X} &\in \mathbb{X}^{H+1} \qquad \textit{State Constraint}\\ \mathbf{U} &\in \mathbb{U}^H \qquad \textit{Input Constraint}\end{aligned} \tag{22}$$

Only the first control action $\boldsymbol{u_1^*}$ It is implemented while the remaining samples are discarded, and the exact steps are repeated by shifting the prediction horizon at each sampling instant k.

- DQ-Net-MPC Problem Formulation:

The proposed DQ-Net-MPC approach is designed to allow a swarm-UAV system to solve the following optimization problem cooperatively:

Consider a swarm of UAVs labeled by $q = \{1,\dots,n\}$ that interact with each other through a communication network represented by a graph $\mathcal{G}$. Each $\mathrm{UAV_i}$ pose and velocity are represented in dual quaternion space by $\underline{\mathbf{x}}_i, \underline{\mathbf{v}}_i \in \mathcal{H}_p$, similarly to [11], let $\underline{\mathbf{x}}_i^d, \underline{\mathbf{v}}_i^d \in \mathcal{H}_p$ is the desired position and velocity of each $\mathrm{UAV_i}$ with respect to the center of the desired formation. The objective is to design a formation tracking controller with minimal position error ϵ. Given $\underline{\mathbf{x}}_i$ and $\underline{\mathbf{x}}_i^d$ for all $\mathrm{UAV_i}$, the dual quaternion error $\underline{\mathbf{x}}_e \in \mathcal{H}_p$ is defined as:

$$\underline{\mathbf{x}}_e = \underline{\mathbf{x}}_i^*\underline{\mathbf{x}}_i^d = \mathbf{r}_e + \varepsilon\frac{1}{2}\mathbf{p}_e\mathbf{r}_e, \tag{23}$$

where $\mathbf{r}_e = \cos(\frac{\emptyset_e}{2}) + \mathbf{n}\sin(\frac{\emptyset_e}{2})$ is the orientation error and $\mathbf{p}_e \in \mathbb{H}$ is the position error. As seen when $\underline{\mathbf{x}}_i \to \underline{\mathbf{x}}_i^d$ then $\underline{\mathbf{x}}_e \to 1$, which in turn implies $\mathbf{p}_e \to 0$ and $\mathbf{r}_e \to 1 \Rightarrow \emptyset_e \to 0$. A workaround to this problem is to define an invariant dual quaternions error function as follows:

$$\underline{\mathbf{e}}_\mathbf{t} = 1 - \underline{\mathbf{x}}_e = 1 - \underline{\mathbf{x}}_i^*\underline{\mathbf{x}}_i^d, \tag{24}$$

Thus, when $\underline{\mathbf{x}}_i \to \underline{\mathbf{x}}_i^d$ then $\underline{\mathbf{e}}_\mathbf{t}$ Converges to 0, yields the swarm-UAV tracks the desired formation. Multiplying both sides of (24) by $(\underline{\mathbf{x}}_i^d)^*$:

$$\underline{\mathbf{x}}_i^* = (\underline{\mathbf{x}}_i^d)^* - \underline{\mathbf{e}}_\mathbf{t}(\underline{\mathbf{x}}_i^d)^*, \tag{25}$$

The time derivative of (24) with substitute (25) is given by (26) as follows:

$$\dot{\underline{\mathbf{e}}}_\mathbf{t} = -(\mathbf{v}_i^d)^*\underline{\mathbf{x}}_i^d + \underline{\mathbf{e}}_\mathbf{t}(\underline{\mathbf{x}}_i^d)^*\underline{\mathbf{v}}_i^d - (\underline{\mathbf{x}}_i^d\underline{\mathbf{v}}_i^d)^*, \tag{26}$$

Using $vec_8(.)$ and Hamilton operators, the discrete-time derivative of $\underline{\mathbf{de}}_{\mathbf{t+1}}$ At time instance $t+1$ is given by (27) as follows:

$$\begin{aligned}vec_8(\underline{\mathbf{de}}_{\mathbf{t+1}}) = \bar{\mathbf{H}}_8((\underline{\mathbf{x}}_i^d)^*\underline{\mathbf{v}}_i^d)vec_8(\underline{\mathbf{e}}_\mathbf{t})\\ - \bar{\mathbf{H}}_8(\underline{\mathbf{x}}_i^d)\mathbf{C}_8 vec_8(\underline{\mathbf{v}}_i)\\ - vec_8(\underline{\mathbf{x}}_i^d\underline{\mathbf{v}}_i^d)^*,\end{aligned} \tag{27}$$

Thus, the compact form of the error model of the swarm-UAV formation tracking is rewritten by (28) as follows:

$$\mathbf{de_{t+1}} = \mathbf{A_t}\mathbf{e_t} + \mathbf{B_t}\mathbf{u_t} + \mathbf{C_t}, \tag{28}$$

where $\mathbf{A_t} = diag\left(\bar{\mathbf{H}}_8((\underline{\mathbf{x}}_i^d)^*\underline{\mathbf{v}}_i^d)\right), \mathbf{B_t} = diag\left(-\bar{\mathbf{H}}_8(\underline{\mathbf{x}}_i^d)\mathbf{C}_8\right), \mathbf{C_t} = -\mathrm{vec}_8(\underline{\mathbf{x}}_i^d\underline{\mathbf{v}}_i^d)^*$, and assuming that $\mathbf{u_t} = \mathbf{v}_i = \boldsymbol{\mathcal{L}}\left(\mathrm{vec}_8(\underline{\mathbf{x}}_i)\right)^T$ Concerning equation (15).

Table 1 describes the proposed DQ-Net-MPC as a formation tracking controller and explains the pseudo code of the DQ-Net-MPC algorithm.

Algorithm 1 DQ-Net-MPC-based Formation Tracking Control

Inputs: UAVs number n; Laplacian matrix $\boldsymbol{\mathcal{L}} \in \mathbb{R}^{n\times n}$; initial states $\underline{\mathbf{x}}_{\mathrm{i}}^{\mathbf{0}} \in \mathcal{H}_{\mathrm{p}}$; the desired trajectory $\underline{\mathbf{x}}_{\mathbf{d}}, \underline{\mathbf{v}}_{\mathbf{d}} \in \mathcal{H}_{\mathrm{p}}$; sample time T_s; prediction horizon H; controller weighing matrices $\mathbf{Q} \in \mathbb{R}^{8n\times 8n}, \mathbf{R} \in \mathbb{R}^{8n\times 8n}$, $\epsilon = 10^{-3}$.
Output: The optimal control $\mathbf{u}_{\mathbf{t}}^{*} \in \mathbb{R}^{8\mathrm{n}}$
Initialization:

Define the CasADi structures and set the *ipopt* optimizer[1] parameter.
for all $\mathrm{UAV_i}$ at time instance t **do**
    Calculate the error $\underline{\mathbf{x}}_{\mathbf{e}}^{\mathbf{t}} = \left(\underline{\mathbf{x}}_{\mathrm{i}}^{\mathbf{0}}\right)^{*}\underline{\mathbf{x}}_{\mathbf{d}}^{\mathbf{t}} \in \mathcal{H}_{\mathrm{p}}$;
    Calculate the invariant dual quaternions error $\mathbf{e_t} = \mathbf{1} - \underline{\mathbf{x}}_{\mathbf{e}}^{\mathbf{t}}$;
    Construct $\mathbf{A_t}, \mathbf{B_t}, \mathbf{C_t}$;
    Define the control input as: $\mathbf{u_t} = \mathbf{v_i} = \boldsymbol{\mathcal{L}}\left(vec_8(\underline{\mathbf{x}}_{\mathrm{i}}^{\mathbf{0}})\right)^{\mathrm{T}}$;
    Define the kinematic error model state equation $\mathbf{de_{t+1}} = \mathbf{A_t e_t} + \mathbf{B_t u_t} + \mathbf{C_t}$;
    Set the states and control inputs constraints $\mathbf{x_{max}}, \mathbf{x_{min}}, \mathbf{u_{max}}, \mathbf{u_{min}}$;
    while ($\|\mathbf{e_t}\|_2 > \epsilon$) **do**
      for $k = 1$ to $H - 1$ **do**
      Using $vec_8(.)$ and Hamilton operators, apply the kinematics error model,
        solve the optimization problem and find the minimizing control sequence

$$\mathrm{u}_{\mathrm{t}}^{*} = \underset{\mathbf{u_t}\in\mathbb{R}^{8N\times H}}{argmin} \sum_{k=0}^{H-1} \mathbf{e}_{\mathbf{t+k}}^{\mathrm{T}}\mathbf{Q}\mathbf{e}_{\mathbf{t+k}} + \mathbf{u}_{\mathbf{t+k}}^{\mathrm{T}}\mathbf{R}\mathbf{u}_{\mathbf{t+k}}$$

This paper used the CasADi package integrated with the MATLAB environment to implement the proposed controller. CasADi offers faster solutions for optimization problems than the MATLAB optimization toolbox and has higher accuracy [13].

IV. Results

This section discusses the proposed control approach's effectiveness and highlights the controller performance with three different scenarios. A simulation study using the CasADi package integrated with the MATLAB environment is carried out to examine the performance of the proposed DQ-Net-MPC approach.

Consider a homogenous swarm with $n = 6$ UAVs labeled by $q = \{1, .., 6\}$. These UAVs work cooperatively in 3D space and can only interact with each other through a local communication network with fixed topology, shown in Figure 3. The network properties are mathematically modeled using an undirected graph $\mathcal{G}$ with matrices $(\boldsymbol{\mathcal{D}}, \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{L}})$.

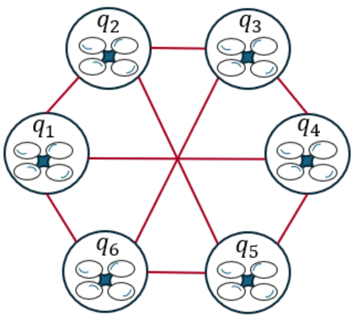


Figure (2): *The graph representation of the swarm-UAV.*

Referring to Figure 3, the graph matrices are written as follows:

$$\boldsymbol{\mathcal{D}} = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix}, \boldsymbol{\mathcal{A}} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}, \quad (29)$$

$$\boldsymbol{\mathcal{L}} = \boldsymbol{\mathcal{D}} - \boldsymbol{\mathcal{A}} = \begin{bmatrix} 3 & -1 & 0 & -1 & 0 & -1 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ 0 & -1 & 3 & -1 & 0 & -1 \\ -1 & 0 & -1 & 3 & -1 & 0 \\ 0 & -1 & 0 & -1 & 3 & -1 \\ -1 & 0 & -1 & 0 & -1 & 3 \end{bmatrix},$$

The constraints considered on the states mainly rely on the communication limits, with a maximum distance for the swarm-UAV. $d_{max} = 50m$, and the kinematics/dynamics with maximum control input relies on the UAV's specifications.

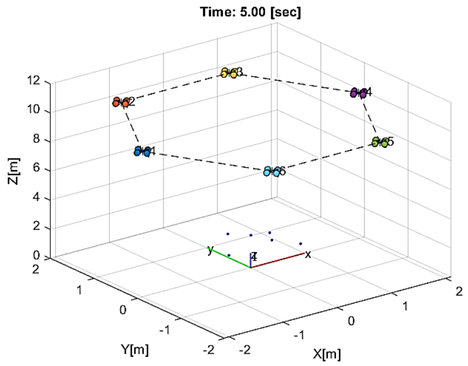
The tuning parameters are chosen as follows: The weighting matrices of the states and control vectors $\mathbf{Q} = 2 * eye(8n, 8n)$, $\mathbf{R} = 0.1 * eye(8n, 8n)$, the sample time $T_s = 0.2\ sec$, and the prediction horizon $H = 15\ samples$. To evaluate the performance of the DQ-Net-MPC approach, three scenarios were simulated as follows:

- Scenario 1: Forming a static hexagonal shape;

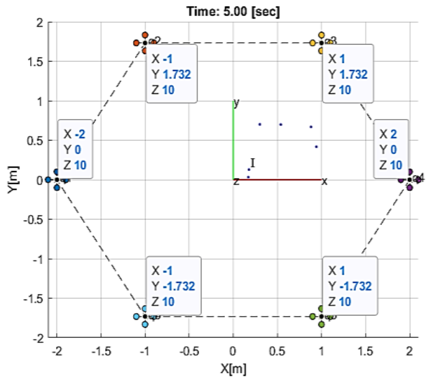
In this scenario, all $\mathrm{UAV_i}$ Have identical performance, and the objective is to maintain a fast vertical motion to a constant altitude $z = 10m$, with a fixed hexagonal formation. $\boldsymbol{\mathcal{F}_d}$. All the desired waypoints, DQ positions, and orientations that define this hexagonal formation are chosen, such as:

$$\boldsymbol{\mathcal{F}_d} = [\mathrm{vec_8}(\underline{\mathbf{x}}_1^{\mathrm{d}}), \mathrm{vec_8}(\underline{\mathbf{x}}_2^{\mathrm{d}}), \dots, \mathrm{vec_8}(\underline{\mathbf{x}}_6^{\mathrm{d}})] = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 1 & 2 & 1 & -1 \\ 0 & 1.732 & 1.732 & 0 & -1.732 & -1.732 \\ 10 & 10 & 10 & 10 & 10 & 10 \end{bmatrix} \quad (30)$$

As seen in Figure 4a-b, the swarm-UAV climbs successfully from random initial positions to the desired formation shape.



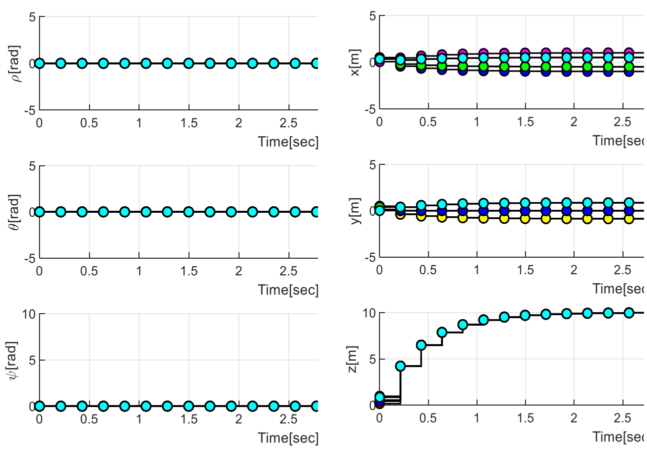
(**a**)



(**b**)

Figure (3): (a) 3D view of swarm desired formation; (b) Top view of the swarm-UAV positions

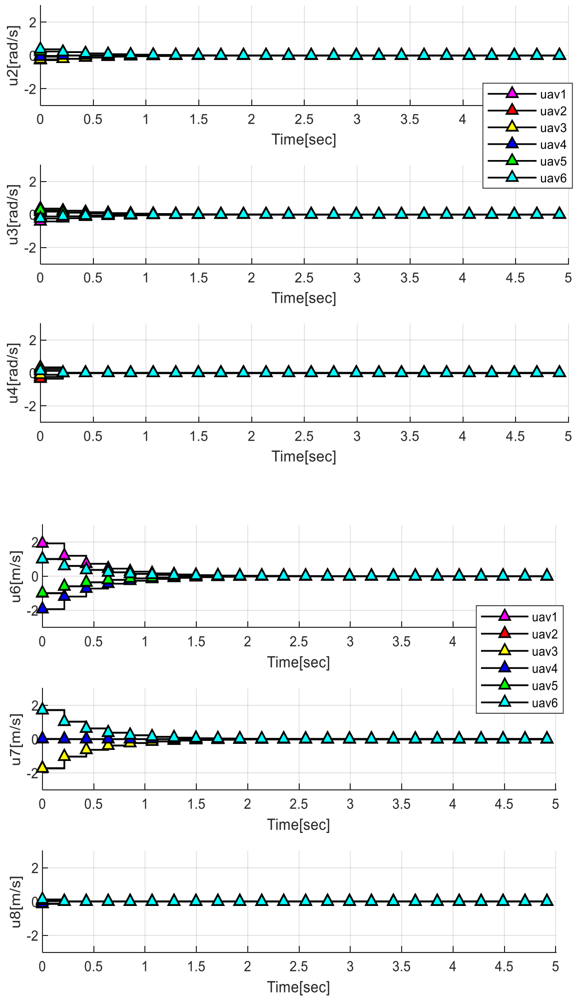
The time evolution of the swarm-UAV orientations and positions performing a fixed formation is shown in Figure 5a-b. It can be observed that the swarm quickly completes the circle formation with an execution time of about 2 sec.



(**a**) (**b**)

Figure (4): Time evolution of the swarm-UAV performing hexagonal formation: (a) The attitudes of all UAVs; (b) The position of all UAVs.

Figure 6a decelerates the calculated attitude control actions. u_2, u_3, u_4 for the orientations roll, pitch, and yaw, respectively. The attitude control of all UAVs indicates the stability of the entire controlled swarm.



a-b

Figure (5): (a) The calculated attitude control actions performing hexagonal formation; (b) The calculated position control actions performing hexagonal formation.

Similarly, Figure 6b decelerates the calculated control actions. u_6, u_7, u_8 for each UAV_i positions $x, y, z,$ respectively. Results illustrate that the proposed controller has no steady-state error, and all UAVs successfully converge to the desired formation shape.

- Scenario 2: Leader- Followers trajectory tracking with a static chain-like formation;

In this scenario, the required task is to move along a square trajectory, where a chain-like formation is adopted for the swarm-UAV and UAV_1 is chosen as the leader.

As seen in Figure 7, the swarm-UAV perfectly tracks a square trajectory with a chain-like formation starting from the leader UAV_1. The followers perform the desired formation attitude and altitude with stability.

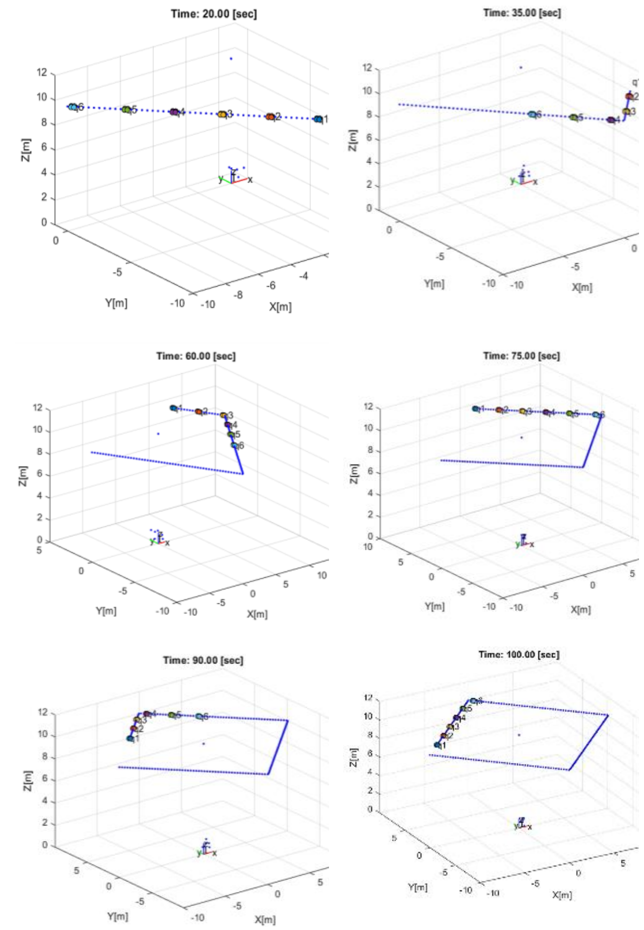


Figure (7): 3D views of the swarm-UAV track a square trajectory with a chain-like formation.

The time evolution of the swarm-UAV coordinates performing a chain-like formation is shown in Figure 8.

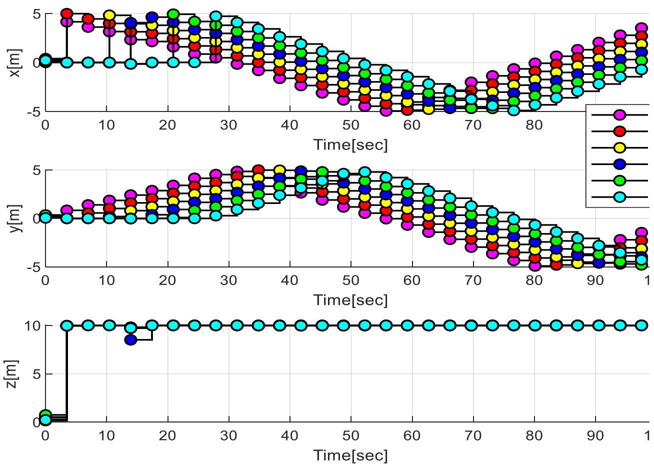
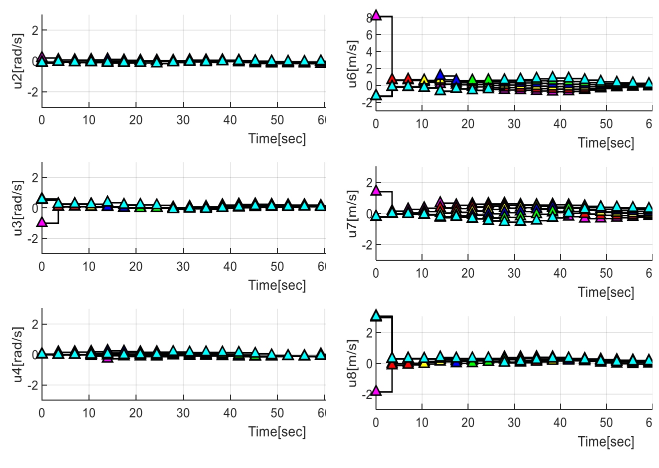


Figure (6): Time evolution of the swarm-UAV tracking a square trajectory with a chain-like formation.

Notice that the swarm achieves the chain-like formation and completes the square trajectory during $100\,sec$. Figure 9a-b illustrates the calculated attitude control actions. u_2, u_3, u_4 and the calculated position control actions u_6, u_7, u_8 for all $\mathrm{UAV_i}$. Notably, DQ-Net-MPC can drive all UAVs to converge smoothly to the desired formation shape.



(a) **(b)**

Figure (7): *(a) The calculated attitude control actions; (b) The calculated position control actions.*

- Scenario 3: Leader- Followers trajectory tracking with a dynamic circular formation.

Here, a circular formation with time-varying diameter is implemented, requiring the swarm to follow the altitude of a virtual leader. q_0 at the center of the formation, as declared in Figure 10.

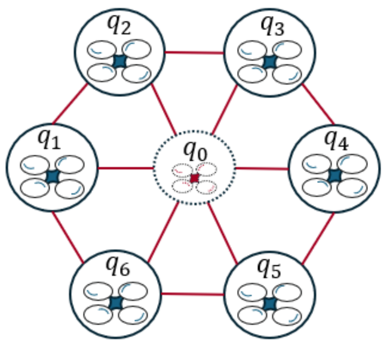


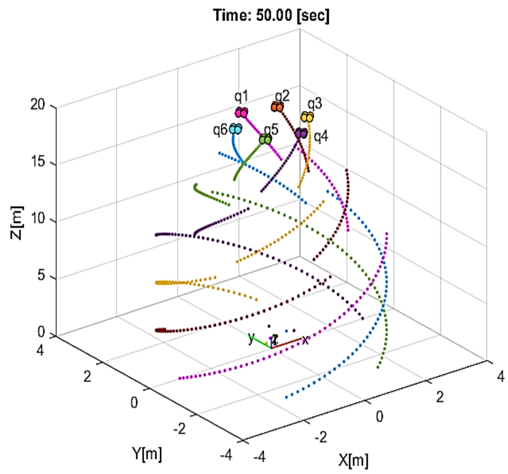
Figure (8): The graph representation of the swarm-UAV with a virtual leader.

The leader q_0 moves with a fixed velocity $v_z = 1m/sec$ about the z-axis. The followers move from their random initial positions to form a circle around $\mathrm{q_0}$ with a time-varying diameter A_t. The followers will turn in that circle and keep moving for all future times. The desired formation $\boldsymbol{\mathcal{F}_d}(\mathrm{t})$ at time instant t with time-varying diameter A_t It is defined as follows:

$$\boldsymbol{\mathcal{F}_d}(\mathrm{t}) = \mathrm{vec_8}\left(\underline{\mathbf{x}}_{\mathrm{i}}^{\mathrm{d}}\right) = \Big[1 \quad 0 \quad 0 \quad 0 \quad 0 \quad \mathrm{A_t cos}\left(0.25\mathrm{t} - \frac{2\pi(\mathrm{i}-1)}{\mathrm{n}}\right) \qquad (31)$$

where the number of UAVs $n = 6$ and $i = 1, \dots, n$ and the attitude angles of all UAVs is $[0 \quad 0 \quad 0]rad$

Figure 11a-b shows the accuracy of the DQ-Net-MPC tracking performance with this scenario. Again, the proposed approach achieves Formation stability, maintaining the desired time-varying diameter.



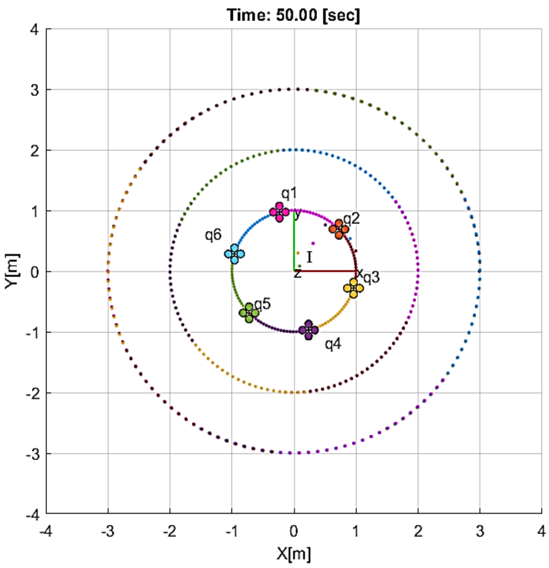


Figure (11): (a) + (b) 3D view of the swarm-UAV tracks a dynamic circular formation; (b) Top view of the swarm-UAV positions.

Figure 12 illustrates the time evolution of the swarm-UAV performing scenario 3. As seen, the swarm-UAV accomplished the trajectory with 50 sec.

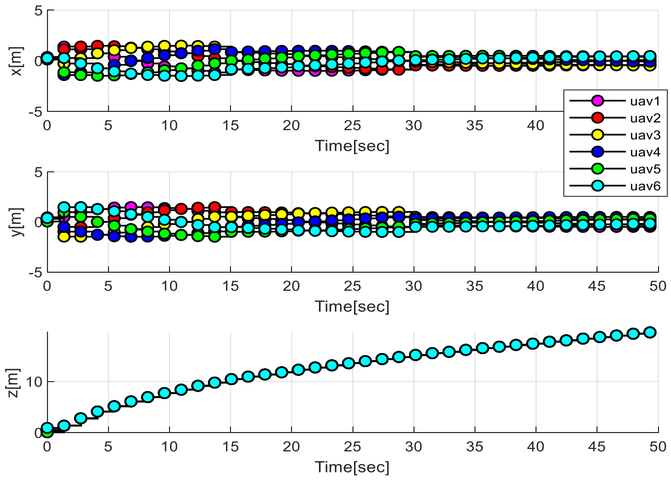
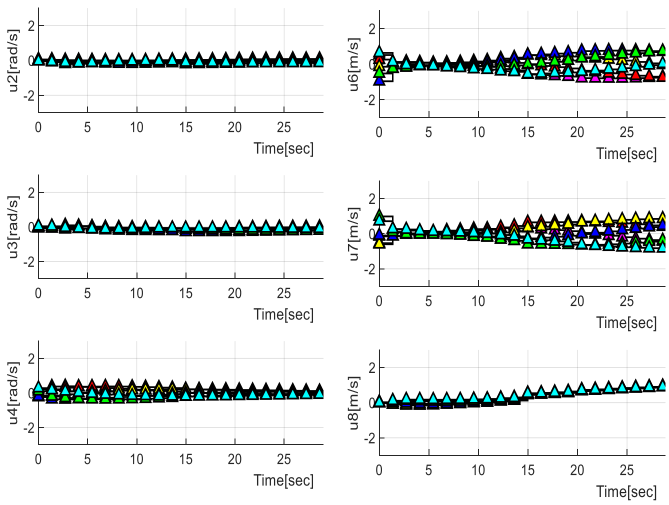


Figure (9**):** Evolution of the swarm-UAV coordinates performing circular dynamic formation.

Finally, the results in Figure 13a-b illustrate the stability of the calculated control actions; the proposed DQ-Net-MPC approach has no steady-state error, and all UAVs successfully converge to the desired formation shape.



(a) **(b)**

Figure (10): (a) The calculated attitude control actions; (b) The calculated position control actions.

Discussion

To highlight the superior performance of the proposed approach, the RMS error is computed for the three scenarios. The most important result from the comparison between the previous scenarios is that the DQ-Net-MPC achieves the minimum tracking error and maintains the desired formations, as shown in Figures 14, 15, and 16.

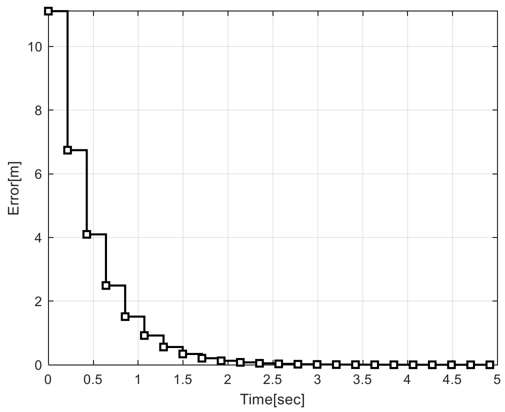


Figure (11): Tracking error norm of scenario 1.

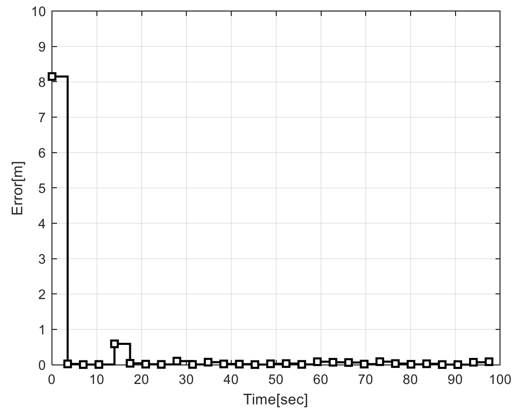


Figure (12): Tracking error norm of scenario 2.

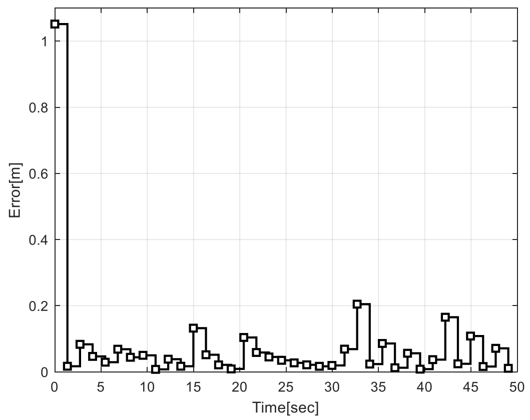


Figure (13): Tracking error norm of scenario 3.

It is worth noting that implementing the proposed DQ-Net-MPC with CasADi optimizer accelerates the system performance, which can be deployed on embedded system units and fast computers for real-world applications.

Table 2 summarizes the results from the three scenarios comparing the achieved minimum errors with the number of iterations, which also support the demonstration of the effectiveness and the stability performance of the proposed DQ-Net-MPC approach.

Table 2. RMS errors of positions and orientations, and the number of iterations of the three tracking scenarios, will be used to validate the effectiveness of the proposed DQ-Net-MPC approach.

	Scenario 1	**Scenario 2**	**Scenario 3**
$\|\mathbf{e_t}\|_2$	$8.95 \times 10^{-5} m$	$4.12 \times 10^{-2} m$	$8.83 \times 10^{-2} m$
Iterations	40	718	916

V. Conclusions

This paper deals with the design and implementation problem of a novel formation tracking controller based on a modified networked model predictive control approach using dual quaternion algebra (DQ-Net-MPC). The proposed approach has been tested for a homogenous, large-scale swarm system consisting of six UAVs working cooperatively to achieve the desired formation shapes.

After a short introduction to DQ algebra, graph theory, and the MPC algorithm, the problem formulation and proposed solution are explained in detail.

To further validate the proposed approach, MATLAB simulation results supported by the CasADi optimization framework have been presented for three scenarios.

The DQ mathematical kinematic modelling of the swarm-UAV enhanced the accuracy of the proposed controller with a simple, compact formula that is easy to use in the future for real-world applications. The results have emphasized fast and accurate convergence to the desired time-varying formation of the interconnected swarm-UAV.

Future work is carried out by testing the performance of DQ-Net-MPC with switched network topology; deploying the proposed approach using Raspberry Pi units supported with ROS (robot operating system); exploring advanced communication technologies (e.g., the Internet of Things, IoT); and enhancing the swarm system with an AI-task controller based on reinforcement learning.

Declaration of Conflicting Interests

The authors declared no potential conflicts of interest.

Funding

The authors received no financial support for this book review.

Human Participants

This paper does not involve human participants.

Originality Note

The authors confirm that the manuscript is their original work, and if others' works are used, they are properly cited/quoted.

VI. REFERENCES

1. Adorno, B. V., & Marinho, M. M. (2021). DQ Robotics: A library for robot modeling and control. IEEE Robotics & Automation Magazine, 28(3), 102–116. https://doi.org/10.1109/MRA.2020.2997920
2. Ajwad, S. A. (2020). Distributed control of multi-agent systems under communication constraints: Application to robotics (Doctoral dissertation, Université de Poitiers). TEL Archives. https://tel.archives-ouvertes.fr/tel-03012285
3. Alonso-Mora, J., Baker, S., & Rus, D. (2017). Multi-robot formation control and object transport in dynamic environments via constrained optimization. The International Journal of Robotics Research, 36(9), 1000–1021. https://doi.org/10.1177/0278364917719333

4. Alsammak, I. L. H., Mahmoud, M. A., Aris, H., AlKilabi, M., & Mahdi, M. N. (2022). The use of swarms of unmanned aerial vehicles in mitigating area coverage challenges of forest-fire-extinguishing activities: A systematic literature review. Forests, 13(5), 811. https://doi.org/10.3390/f13050811
5. Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J. B., & Diehl, M. (2019). CasADi: A software framework for nonlinear optimization and optimal control. Mathematical Programming Computation, 11(1), 1–36. https://doi.org/10.1007/s12532-018-0139-4
6. Asaamoning, G., Mendes, P., Rosário, D., & Cerqueira, E. (2021). Drone swarms are networked control systems that integrate networking and computing. [Conference presentation or manuscript].
7. Aspragkathos, S. N., Rousseas, P., Karras, G. C., & Kyriakopoulos, K. J. (2024). Multirotor nonlinear model predictive control is based on the visual servicing of evolving features. arXiv. https://doi.org/10.48550/arXiv.2409.16665
8. Azari, M. M. (2019). Wireless communications with unmanned aerial vehicles (UAVs) (Doctoral dissertation, Faculty of Engineering Science, Belgium).
9. Borghi, G., & Herty, M. (2023). Model predictive control strategies using consensus-based optimization. arXiv. https://doi.org/10.48550/arXiv.2312.13085
10. Chang, C.-W., & Shiau, J.-K. (2018). Quadrotor formation strategies based on distributed consensus and model predictive controls. Applied Sciences, 8(11), 2246. https://doi.org/10.3390/app8112246
11. Chitikena, H., Sanfilippo, F., & Ma, S. (2023). Robotics in search and rescue (SAR) operations: An ethical and design perspective framework for the response phase. Applied Sciences, 13(3), 1800. https://doi.org/10.3390/app13031800
12. Dubois, L., & Suzuki, S. (2018). Formation control of multiple quadcopters using model predictive control. Advanced Robotics, 32(19), 1037–1046. https://doi.org/10.1080/01691864.2018.1470572
13. Ebel, H. (2021). Distributed control and organization of communicating mobile robots: Design, simulation, and experimentation. Shaker Verlag.
14. Elhesasy, M., et al. (2023). Non-linear model predictive control using the CasADi package for trajectory tracking of a quadrotor. Energies, 16(5), 2143. https://doi.org/10.3390/en16052143
15. Elwin, M. L., Strong, B., Freeman, R. A., & Lynch, K. M. (2023). Human-multirobot collaborative mobile manipulation: The omnid mocobots. IEEE Robotics and Automation Letters, 8(1), 376–383. https://doi.org/10.1109/LRA.2022.3226366
16. Figueredo, L. F. D. C. (2016). Kinematic control based on dual quaternion algebra and its application to robot manipulators (Doctoral dissertation, Universidade de Brasília). https://doi.org/10.26512/2016.07.T.21834
17. Ghaderi, F., Toloei, A., & Ghasemi, R. (2024). Formation control and obstacle avoidance of a multi-quadrotor system based on model predictive control and improved artificial potential field. International Journal of Engineering, 37(1), 115–126. https://doi.org/10.5829/ije.2024.37.01a.11
18. Giribet, J. I., Colombo, L. J., Moreno, P., Mas, I., & Dimarogonas, D. V. (2021). Dual quaternion cluster-space formation control. IEEE Robotics and Automation Letters, 6(4), 6789–6796. https://doi.org/10.1109/LRA.2021.3095037
19. Gunn, T., & Anderson, J. (2015). Dynamic heterogeneous team formation for robotic urban search and rescue. Journal of Computer and System Sciences, 81(3), 553–567. https://doi.org/10.1016/j.jcss.2014.11.009
20. Hartmann, V. N. (2023). Scalable multi-agent task and motion planning for robotic building construction. [Unpublished manuscript].
21. Hirata-Acosta, J., Pliego-Jiménez, J., Cruz-Hernádez, C., & Martínez-Clark, R. (2021). Leader–follower formation control of wheeled mobile robots without attitude measurements. Applied Sciences, 11(12), 5639. https://doi.org/10.3390/app11125639
22. Iskender, O. B., et al. (2019). Dual quaternion-based autonomous rendezvous and docking via model predictive control. [Conference presentation or manuscript].
23. Joukhadar, A., AlChehabi, M., & Jejeh, A. (2019). Advanced UAVs nonlinear control systems and applications. Unmanned Robotic Systems and Applications, 79.
24. Joukhadar, A., AlChehabi, M., Stöger, C., & Müller, A. (2019). Trajectory tracking control of a quadcopter UAV using nonlinear control. In R. Rizk & M. Awad (Eds.), Mechanism, machine, robotics and mechatronics sciences (pp. 271–285). Springer International Publishing. https://doi.org/10.1007/978-3-319-89911-4_20
25. Joukhadar, A., Hanna, D. K., & Abo Al-Izam, E. (2020). UKF-based image filtering and 3D reconstruction. In O. Sergiyenko, W. Flores-Fuentes, & P. Mercorelli (Eds.), Machine vision and navigation (pp. 267–289). Springer International Publishing. https://doi.org/10.1007/978-3-030-22587-2_9
26. Joukhadar, A., Hanna, D. K., Müller, A., & Stöger, C. (2019). UKF-assisted SLAM for 4WDDMR localization and mapping. In R. Rizk & M. Awad (Eds.), Mechanism, machine, robotics and mechatronics sciences (Vol. 58, pp. 259–270). Springer International Publishing. https://doi.org/10.1007/978-3-319-89911-4_19
27. Khan, U. A., & Moura, J. M. F. (2007). Model distribution for distributed Kalman filters: A graph theoretic approach. In 2007 Conference Record of the Forty-First Asilomar Conference on Signals, Systems and Computers (pp. 611–615). IEEE. https://doi.org/10.1109/ACSSC.2007.4487286
28. Li, Y., Liu, M., Lian, J., & Guo, Y. (2022). Collaborative optimal formation control for heterogeneous multi-agent systems. Entropy, 24(10), 1440. https://doi.org/10.3390/e24101440
29. Liu, H., Meng, Q., Peng, F., & Lewis, F. L. (2020). Heterogeneous formation control of multiple UAVs with limited-input leader via reinforcement learning. Neurocomputing, 412, 63–71. https://doi.org/10.1016/j.neucom.2020.06.040
30. Liu, Y., Dong, X., Shi, P., Liu, J., & Zeng, Q. (2022). Integrated active fault-tolerant formation control for networked multi-agent systems. IET Control Theory

& Applications, 16(7), 718–728. https://doi.org/10.1049/cth2.12262

31. Loria, A., Dasdemir, J., & Alvarez Jarquin, N. (2016). Leader–follower formation and tracking control of mobile robots along straight paths. IEEE Transactions on Control Systems Technology, 24(2), 727–732. https://doi.org/10.1109/TCST.2015.2437328
32. Lunze, J. (Ed.). (2014). Control theory of digitally networked dynamic systems. Springer International Publishing. https://doi.org/10.1007/978-3-319-01131-8
33. Mas, I., & Kitts, C. (2017). Quaternions and dual quaternions: Singularity-free multirobot formation control. Journal of Intelligent & Robotic Systems, 87(3–4), 643–660. https://doi.org/10.1007/s10846-016-0445-x
34. Miller, O. G., & Gandhi, V. (2021). A survey of modern exogenous fault detection and diagnosis methods for swarm robotics. Journal of King Saud University – Engineering Sciences, 33(1), 43–53. https://doi.org/10.1016/j.jksues.2019.12.005
35. Nekoo, S. R., Á. Acosta, J., & Ollero, A. (2022). Quaternion-based state-dependent differential Riccati equation for quadrotor drones: Regulation control problem in aerobatic flight. Robotica, 40(9), 3120–3135. https://doi.org/10.1017/S0263574722000091
36. Perez-Saura, D., Fernandez-Cortizas, M., Perez-Segui, R., Arias-Perez, P., & Campoy, P. (2023). Urban firefighting drones: Precise throwing from UAV. Journal of Intelligent & Robotic Systems, 108(4), 66. https://doi.org/10.1007/s10846-023-01883-6
37. Qin, B., Zhang, D., Tang, S., & Wang, M. (2022). Distributed grouping cooperative dynamic task assignment method of UAV swarm. Applied Sciences, 12(6), 2865. https://doi.org/10.3390/app12062865
38. Savino, H. J., Pimenta, L. C. A., Shah, J. A., & Adorno, B. V. (2019). Pose consensus based on dual quaternion algebra with application to decentralized formation control of mobile manipulators. arXiv. https://doi.org/10.48550/arXiv.1810.08871
39. Seisa, A. S., Satpute, S. G., Lindqvist, B., & Nikolakopoulos, G. (2022). An edge-based architecture for offloading model predictive control for UAVs. Robotics, 11(4), 80. https://doi.org/10.3390/robotics11040080
40. Sial, M. B., et al. (2022). Bearing-based distributed formation control of unmanned aerial vehicle swarm by quaternion-based attitude synchronization in three-dimensional space. Drones, 6(9), 227. https://doi.org/10.3390/drones6090227
41. Siciliano, B., Uchiyama, M., Khatib, O., Fraisse, P., Poignet, P., & Perdereau, V. (2011). Two-arm manipulation: From manipulators to enhanced human–robot collaboration (Doctoral dissertation, Université Montpellier II).
42. Stolfi, D. H., & Danoy, G. (2022). Optimising autonomous robot swarm parameters for stable formation design. In Proceedings of the Genetic and Evolutionary Computation Conference (pp. 1281–1289). ACM. https://doi.org/10.1145/3512290.3528709
43. Stomberg, G., Ebel, H., Faulwasser, T., & Eberhard, P. (2023). Cooperative distributed MPC via decentralized real-time optimization: Implementation results for robot formations. arXiv. https://doi.org/10.48550/arXiv.2301.07960
44. Valverde, A., & Tsiotras, P. (2018). Spacecraft robot kinematics using dual quaternions. Robotics, 7(4), 64. https://doi.org/10.3390/robotics7040064
45. Xuan-Mung, N., & Hong, S. K. (2019). Robust adaptive formation control of quadcopters based on a leader–follower approach. International Journal of Advanced Robotic Systems, 16(4), 172988141986273. https://doi.org/10.1177/1729881419862733
46. Yang, S., Li, T., Shi, Q., Bai, W., & Wu, Y. (2020). Artificial potential-based formation control with collision and obstacle avoidance for second-order multi-agent systems. In 2020, the 7th International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS) (pp. 58–63). IEEE. https://doi.org/10.1109/ICCSS52145.2020.9336836
47. Yang, W., & Ye, X. (2024). Distributed formation fault tolerance control based on an improved artificial potential field method. In 2024 IEEE International Conference on Mechatronics and Automation (ICMA) (pp. 1628–1633). IEEE. https://doi.org/10.1109/ICMA61710.2024.10633187
48. Zaouche, L. (2017). Secure and reliable communications in swarm networks (Doctoral dissertation, Université de Technologie de Compiègne). TEL Archives. https://tel.archives-ouvertes.fr/tel-01707900
49. Zhuang, H., Dong, K., Wang, N., & Dong, L. (2022). Multi-robot leader grouping consistent formation control method research with low convergence time based on nonholonomic constraints. Applied Sciences, 12(5), 2300. https://doi.org/10.3390/app12052300

About the authors

Dr. Abdulkader Joukhadar has been a full chair Professor of Intelligent Control Systems in the Department of Mechatronics Engineering, University of Aleppo, since 2004. In 1992, he earned his BSc(Honors) first class in Electrical and Electronic Engineering; in 1999, he got his MPhil in Industrial Electronics and Control from the University of Birmingham, UK. In 1999, he joined the research group of Intelligent Motion Control Systems and Condition Monitoring at the University of Aberdeen in 2004. He got his PhD focusing on Intelligent Control Systems from the same university. With over two decades of experience in academia and research, Professor Joukhadar has significantly contributed to mechatronics, particularly in intelligent control systems and robotics, Unscented Kalman Filter, and Extended Kalman Filter. He is the head of the Integrated Mechatronics Systems Research Group and the director of the Intelligent Robotics Systems Lab at the University of Aleppo. His research interests include autonomous systems, robotics and robot control, Kalman Filtering, Swarm Networking, and the development of innovative control strategies for nonlinear control systems. Professor Joukhadar has published numerous papers in prestigious journals and international conferences and actively collaborates with international researchers to advance the Robotics and Intelligent Systems field.

Prof. Dalia Kass Hanna has been a member of the engineering technical staff in the Department of Mechatronics Engineering since 2012. She obtained her BSc in Automation and Control from the Faculty of Electrical and Electronic Engineering, University of Aleppo. She got her MSc in Automation and Control from the University of Aleppo. Presently, she is a PhD candidate, and her main topic is Swarm Networking Robotics Systems.

Dr. Basem Fares has been an associate Professor of Robust and Advanced Optimal Control in the Department of Control Engineering and Automation, University of Aleppo, since 2001. In 1991, he earned his BSc(Honors) with first-class honors in Electrical and Electronic Engineering honors. In 1999, he got his DEA in Automatic Control Engineering from Poly-Technique Institute-Grenoble, France. In 2001, He got his PhD from the University of Toulouse III, France, focusing on Robust Control and Advanced Optimal Control Methodologies. With more than 25 years of experience in academia and research, Dr Fares has made vital contributions to robust and advanced optimal control. His research interests include nonlinear control systems, robust control systems, and Kalman Filtering. He has published numerous papers in prestigious journals and international conferences and actively collaborates with international researchers to advance the field of Robust Control Systems.