Path Planning and Formation Control for UAV-Enabled Mobile Edge Computing Network

Kheireddine Choutri, Mohand Lagha, Souham Meshoul, Samiha Fadloun

Abstract: Recent developments in unmanned aerial vehicles (UAVs) have led to the introduction of a wide variety of innovative applications, especially in the Mobile Edge Computing (MEC) field. UAV swarms are suggested as a promising solution to cope with the issues that may arise when connecting Internet of Things (IoT) applications to a fog platform. We are interested in a crucial aspect of designing a swarm of UAVs in this work, which is the coordination of swarm agents in complicated and unknown environments. Centralized leader-follower formations are one of the most prevalent architectural designs in the literature. In the event of a failed leader, however, the entire mission is canceled. This paper proposes a framework to enable the use of UAVs under different MEC architectures, overcomes the drawbacks of centralized architectures, and improves their overall performance. The most significant contribution of this research is the combination of distributed formation control, online leader election, and collaborative obstacle avoidance. For the initial phase, the optimal path between departure and arrival points is generated, avoiding obstacles and agent collisions. Next, a quaternion-based sliding mode controller is designed for formation control and trajectory tracking. Moreover, in the event of a failed leader, the leader election phase allows agents to select the most qualified leader for the formation. Multiple possible scenarios simulating real-time applications are used to evaluate the framework. The obtained results demonstrate the capability of UAVs to adapt to different MEC architectures under different constraints. Lastly, a comparison is made with existing structures to demonstrate the effectiveness, safety, and durability of the designed framework.

Keywords: Mobile Edge Computing; UAV; formation control; leader election; path planning; obstacle avoidance; artificial intelligence; multi-agent systems

1. Introduction

The majority of Internet of Things (IoT) applications require a number of computational tasks that cannot be performed locally on IoT devices due to the vast amount of data that are generated and need to be processed in real time. Because of their greater processing and storage capacity, external devices in centralized servers in remote clouds or at the edge are used to perform these IoT tasks, which require offloading operations. Connecting an Internet of Things network to a Mobile Edge Computing (MEC) platform, on the other hand, could be a challenging endeavor. The use of unmanned aerial vehicles (UAVs) is one possible solution to this issue. A survey concerned UAV-Enabled Mobile Edge Computing for IoT Devices can be found on [1]. Depending on their role in the network, UAVs can be used as fog nodes [2] or as relays to improve the connectivity of a wireless network [3]. In [4], a swarm of UAVs is employed in the fog layer to detect objects in real time videos. UAVs with onboard computing resources can provide offloading services to nearby mobile users (MUs). Within the constraints of the available computing power, numerous
strategies were proposed to reduce the processing time among MUs. While [5] offers a compute offloading strategy based on Deep Deterministic Policy Gradient (DDPG), Ref. [6] uses Reinforcement Learning (RL) to enhance the Quality-of-Service for each terminal user. Depending on the size and weight of the UAV, work offloading in UAV-enhanced edges can be challenging. Due to the influence that UAV-reliant networks may have on QoS and performance, it is vital that they be designed with energy efficiency in mind. The authors of [7] offer an approach that simultaneously optimizes the offloading of tasks and the UAV’s flying path. Two objective functions addressing energy-efficient offloading and safe path planning are proposed in [8] as part of a constrained multi-objective optimization problem for UAVs. The authors of [9] are primarily concerned with path planning under the paradigm of edge computing. In addition, the authors of [10] present an algorithm for creating a feasible task migration path, taking into account the migration distance between UAVs, the load condition of UAVs, and environmental characteristics. The authors of [11] describe a cloud-based motion planning method for coordinating the safe movement of a large number of mobile robots.

The mobility of UAVs and their ever-increasing computational power paved the path for their usage in a wide variety of novel applications. Recently, a significant amount of work and attention has been focused on their application in the support of mobile edge computing systems. Several different MEC systems that are aided by UAVs have been proposed [12]. On the other hand, it has been reported that a single UAV-based solution might not be adequate to complete the mission due to the limited life of its battery and the limited computing power it possesses. A swarm of drones has many advantages over a single drone, including increased reliability and decreased operation time. As a consequence, there has been a surge in the use of swarms of drones rather than a single drone. Dealing with a swarm of drones gives rise to a number of challenges that need to be addressed among which are multi-agent systems, path planning, and formation control.

Multi-agent systems have received much attention from the scientific community in recent years. Multiple collaborating UAVs can complete more challenging jobs and achieve more complicated objectives than a single UAV. Various techniques and designs have been proposed in the literature, including behavior-based [13], virtual structure [14], potential field [15], and leader–follower [16–18]. In the centralized leader–follower (L-F) scenario, a “leader” agent has the reference motion that the other agents, “followers”, track it. To facilitate collaboration, the leader conveys its state to the followers via an appropriate communication channel or link. Consequently, a leader’s failure will result in mission failure. The leader election is one of the recently researched challenges in leader–follower settings. In most cases, the leader is assumed to be a specific agent picked at the start of the job; this event is known as a static leader election. Ref. [19] offers a new leader election approach based on an adaptive/reliable network structure. Ref. [20] offers a distributed leader election method that does not require direct inter-agent communication. For online leader election, the authors of [21] suggest a fully-decentralized adaptive technique capable of selecting the best leader among the existing leader’s neighbors on a regular basis. This author describes a distributed online leader election model. The swarm agents are adaptive to unexpected leader failure and changes in network topology.

In formation control, the motion of UAVs is tightly restricted to preserve the formation topology. Consensus algorithms have been explored extensively in the literature for this purpose [22,23]. Numerous control mechanisms were proposed for UAV formation control, with the underlying theory described in [24]. Ref. [25] suggests a second-order consensus algorithm to follow a specified external reference in their work, whereas [26] frames the formation control problem as a position control problem to be solved. In contrast, spacecraft formation vehicles use a mechanism for robust attitude control, as described in [23]. The prior control strategies were able to sustain the formation, requiring a position estimate for both the leader and the followers. In addition, communication limitations pertaining to the proposed typologies were not considered. A consensus-based attitude
A formation controller is utilized for this study. The formation topology is then preserved with minimal data interchange, and the controller is resilient to external disturbances.

Avoiding other UAVs and natural obstacles is essential for mission success and safe operation. There have been numerous proposed solutions to this problem. The avoidance of obstacles by unmanned aerial vehicles is examined in [27]. In [28], the methods of conflict identification and resolution for a multi-UAV collision-avoidance system are investigated. Therefore, a modified technique is given for tentacle construction and collision avoidance for multiple UAVs operating in unstructured environments. For use in the actual world without the assistance of a pilot, Ref. [29] developed a system that can automatically navigate and avoid obstacles. Moreover, Ref. [30] describes a decentralized control system based on behavior, which employs a sliding mode controller and artificial potential functions. Although other works have addressed the issue of avoiding obstacles when flying in a UAV formation, nobody has yet optimized the resulting trajectory.

This research offers a new paradigm for multi UAVs, one that makes distributed formation control operable across a wide range of MEC architectures, unlike previous work in [31–34]. Therefore, algorithms for optimal trajectory tracking, obstacle avoidance, and leader selection were incorporated into the resulting framework. The suggested framework removes the problems associated with conventional leader–follower development and produces better results. The following are the foundational requirements of the method we propose:

- **R1:** *Multi-agent systems (MAS):* Multi-agent systems are computational systems composed of a large number of interacting computer elements known as agents. To take advantage of the decentralized structure of the multi-agent system, the agent must be provided with some autonomy. When we state that an agent is autonomous, we infer that it can collect data by interacting with other agents and its environment, and then make decisions based on this information.

- **R2:** *Formation control:* According to the proposed method, swarms have a certain topology, such as a rectangle, a diamond, etc. This allows the formation topology to be maintained with little data sharing and makes the controller more resilient to external disturbances.

- **R3:** *Leader election:* In centralized formations, there is often one leader of the swarm; nevertheless, if a single error happens, the entire mission is aborted. In this situation, the other agents must select a new leader.

- **R4:** *Obstacle avoidance:* In centralized formations, there is often one leader of the swarm; nevertheless, if a single error happens, the entire mission is aborted. In this situation, the other agents must select a new leader.

The remainder is organized as follows: In Section 2, the system is introduced with a brief overview of graph theory and MEC architectures. The problem of path planning and obstacle avoidance is formulated in Section 3. Section 4 is dedicated to formation control. The formalization of the dynamic model using quaternions is described first. Then, a suitable SMC controller is developed to preserve the topology of the formation. Finally, the formation transformation for leader election is described. Section 5 examines the simulation results, proposing numerous scenarios and comparing performance. Section 6 presents conclusions and future potential work.

2. Preliminaries

2.1. Mec Architectures

Depending on the role of UAVs in the network, three distinct MEC topologies can be distinguished. In each scenario, the MEC server and MUs are the primary components, which are modeled as data analysis centers located closer to the users on the network’s edge. Using wireless networking and accessible communication technologies, links are built between the different users and servers. Internet is then used to link the MEC server to cloud-based data analysis centers [1].
• **Assisted MEC:** This architecture is often used to provide services following a natural disaster or bombardment-related infrastructure damage. As depicted in Figure 1, the UAV assists mobile users by acting as an aerial MEC server-enabled base station. Each user transfers its computationally intensive activities to one or many UAVs for processing. Therefore, UAVs with long-lasting batteries and powerful CPUs are necessary for this architecture. In addition, this architecture is typically employed to satisfy QoS requirements by optimizing the overall energy consumed by the MUs.

![Figure 1. Assisted MEC architecture.](image)

• **Cellular-Connected MEC:** Figure 2 illustrates these kind of architectures. During a mission, UAVs are viewed as aerial users with computationally intensive tasks, such as path planning and data analysis. Due to limited onboard processing capability, UAVs offload computation to an MEC server on a Ground Base Station (GBSs). In comparison to the previous architecture, the UAVs deployed in this manner have limited batteries and possessors, but they must conduct intense computation tasks.

![Figure 2. Cellular-Connected MEC architecture.](image)
- **Relayed MEC**: As shown in Figure 3, the UAVs in this final architecture serve as relays to help the MUs in offloading their intensive computation tasks to the MEC server of the GBSs. Therefore, none of the UAVs include an MEC server. This architecture is intended to enable long-distance communication links between the MUs and the MEC server in the event that other regular links are interrupted.

![Figure 3. Relayed MEC architecture.](image)

2.2. **System Presentation**

The proposed framework consists of a swarm of UAVs assisting different MEC architectures in a distributed formation. We assume that only the leader is aware of the mission task and all possible formation types in the L-F approach. Followers are expected to collect data from other UAVs and the surrounding environment. The leader takes decisions on specific movements, planned trajectory, and allocation of tasks. The framework is constructed according to the flow chart shown in Figure 4.

![Figure 4. Proposed framework for UAVs swarm monitoring under distributed L-F formations.](image)
The mission is first planned by the user, who specifies the departure and arrival points, the number of UAVs, the formation topology, and the employed MEC architecture. The path planning stage then generates the best path between the departure and arrival points. The control space is optimized, while some constraints in the output and state spaces are applied. This includes physical constraints, actuator limitations, and environment obstacles. Next, the formation control phase aims to preserve the formation’s topology and accommodate any topological variations caused by constraints. Both the leader and the followers have an autopilot system for attitude stabilization and trajectory tracking. As a result, the leader receives the required position to control command for the autopilot in order to arrive at the desired location. The main objective of the UAVs is to offload their computation tasks to/from the MEC server depending on the selected architecture. However, if a failure happens during the mission, the framework includes a safety stage to cope with it. If the leader is unable to complete the task, the followers elect a new leader. The technique seeks to select the most qualified UAV for leadership and switch topologies based on the number of remaining UAVs. Furthermore, if an external obstacle is detected, the leader instructs the followers to split the topology, avoid the obstacle, and return to the original topology.

In this study, we represent the network topology of the UAV swarm using graph theory notations. Nodes in an undirected graph represent network communication nodes, while edges represent communication links between nodes. In the case of UAV swarms, nodes represent individual UAVs and edges represent inter-UAV links, such as wireless communications or sensors. We assume that all communication links are bidirectional. Abstractly, an undirected graph \( G = (v, e) \) is defined by a node set \( v \) with cardinality \( n \), the number of nodes in the graph, and an edge set \( E \) comprised of pairs of nodes, where nodes \( v_i \) and \( v_j \) are adjacent if \( \{v_i, v_j\} \in e \subseteq [v]^2 \). One special family of undirected graphs are tree graphs, denoted by the set \( \Gamma \), where all two-node pairs are connected by exactly one simple path, that is, a connected graph without cycles. A spanning tree of a graph is a tree subgraph that connects all vertices in the graph.

The neighborhood set \( N(v_i) \) of a node \( v_i \) is composed of the set of nodes adjacent to \( v_i \). The scalar \( d = (v_i, v_j) \) is the minimum path length, induced by the graph \( G \), between nodes \( v_i \) and \( v_j \). The adjacency matrix of \( G \) is a positive matrix denoted by \( G^A = \{\omega^A_{ij}\} \in R^{mn} \), where \( \omega^A_{ij} \) represents the entry of the \( i \)-th row and \( j \)-th column of matrix \( G^A \) with \( \omega^A_{ij} = 1 \) if \( (i, j) \in E \), and \( \omega^A_{ij} = 0 \), otherwise. The degree \( \delta_i \) of node \( v_i \) is the number of adjacent nodes. The degree matrix \( G^D \) is a diagonal matrix with \( \delta_i \) at matrix element \( (i, i) \). We also define the diagonal matrix \( G^L = \text{diag}\{\omega^L_1, \ldots, \omega^L_n\} \) representing the status of the agent—if \( \omega^L_i = 1 \), then agent \( i \) is a leader. Otherwise, if \( \omega^L_i = 0 \), then agent \( i \) is a follower.

The Laplacian matrix graph is defined as \( L = G^D - G^A \in R^{mn} \). It plays an important role in the dynamics of the network. An important feature of this matrix is that it is a (symmetric) positive semi-definite matrix. The spectrum is ordered as \( 0 = \lambda_1(L) \leq \lambda_2(L) \leq \ldots \leq \lambda_n(L) \). The interaction positive matrix \( G \) for the L-F formation is defined as follows:

\[
G = G^D - G^A + G^L = L + G^L
\]  

(1)

Based on graph theory, the L-F consensus’ desired trajectory for each agent can be given by:

\[
\begin{align*}
\bar{x}^L_i &= \frac{1}{|N_i|} \sum_{j \in N_i} \left( (x_j + d_{ij}) \right) \quad \text{i follower} \\
\bar{x}^L_i &= \frac{1}{|N_i|} \left( \sum_{j \in N_i} \left( (x_j + d_{ij}) + r + d_{io} \right) \right) \quad \text{i leader}
\end{align*}
\]  

(2)

where \( d_{ij} = d_{io} - d_{io} \) is the inter-distance. Once an observation is available, Equation (2) is reformulated based on the normalized interaction matrix \( \hat{G} \) as follows:
\[
\begin{bmatrix}
  x_i - x_d^i \\
  \vdots \\
  x_n - x_d^n
\end{bmatrix} = (\tilde{G} \otimes I_2) \begin{bmatrix}
  x_i - x_d^i \\
  \vdots \\
  x_n - x_d^n
\end{bmatrix}
\]  

(3)

For this study, a normalized interaction matrix is considered for L-F formations as indicated in Equation (4):

\[
\tilde{G} = (G^D + G^L)^{-1} \cdot G
\]

(4)

Extra edges can be added to the proposed topology as shown in Figure 5. The added two edges and the same vertices constitute the graph \( G_e = (\nu, (\epsilon)^e) \). Then, \( G_e \) and \( \tilde{G} \) yield:

\[
\tilde{G} = G + G_e
\]

(5)

Figure 5. Leader–follower formation topology.

3. Path Planning Stage

The main objective of the path planning stage is to generate the optimal path between the departure and arrival points. A collision avoidance system should often be incorporated into the control architecture to protect UAVs from causing damage to themselves and surrounding objects (other vehicles, persons, and infrastructures). Thus, obstacle avoidance techniques are used to tackle problems with trajectory optimization challenges. The majority of issues in the literature are addressed using dynamic programming, with trajectory optimization occurring throughout the challenge. In this study, however, the obstacle avoidance problem is seen as a Constrained Optimization (CM) problem, with the objective of optimizing the problem’s key parameters: the UAVs desired inter-distance \( (d_d) \) and the radius of the obstacle avoidance zone \( (R_{max}) \).

Next, an energy-aware communication and computation resource is incorporated for the aim of minimizing the total energy consumption. First, the energy consumed by MUs and UAVs while processing tasks is included in the task processing energy consumption. According to capacitance theory [10], the processing energy consumption of an MU or UAV is mostly governed by the CPU performance of the electronic device. Secondly, the communication energy consumption is made up of the energy consumed by the MU’s offload task, as well as the energy consumed by switching topologies between UAVs. Finally, the problem of minimizing the energy consumption by engines and electronic devices is also considered. To summarize, the total energy consumption of all UAVs in the system during a time step \( t \) can be defined by Equation (6):

\[
E_{UAV}^t = \sum_{n \in N} E_{Com,n}^t + E_{Tran,n}^t + E_{Fly,n}^t
\]

(6)
where:

- $E_{Com}^t$ is the energy consumption for processing offloaded tasks of UAV $n$;
- $E_{Tran}^m$ is the transmission energy consumption of MU $m$ offload tasks to UAV $n$;
- $E_{Fly}^n$ is the flight energy consumption of UAV $n$.

The optimization procedure is guided by an aggregation of two objective functions, namely the minimal distance traveled ($e_p$) and the minimum energy ($E$). Consequently, the objective of this work is to minimize traveled distance and energy consumption, while avoiding obstacles and collisions. This can be formulated as follows:

$$\min \Phi = \int_0^T (e_p^T \cdot W \cdot e_p + E^T \cdot Q \cdot E) dt$$

subject to:

- $E \leq E_{\text{max}}$
- $R \geq R_{\text{max}}$
- $d = \bar{d}$

where $W$ and $Q$ are the weighting matrix. Traditional methods can be used to solve the problem. Karush–Kuhn–Tucker conditions are utilized because of inequality limitations.

4. Formation Control Stage

4.1. UAV Model

Quadrotors are a subset of Vertical Take-off and Landing (VTOL) UAVs. Their ability to hover, move forward, and perform vertical takeoffs and landings make them nonlinear and under-actuated systems. This work supposes an attitude representation under quaternions approach to avoid singularities presented in Euler angle presentation. Let the unit quaternion $q(t) \in \mathbb{H}, \bar{q}(t) \in \mathbb{R}^3, q_0(t) \in \mathbb{R}$ be defined with:

$$q(t) = q_0(t) + \bar{q}(t) = q_0(t) + [q_1(t), q_2(t), q_3(t)]^T$$

with $q_0(t)$ a given quaternion and $\bar{q}(t)$ the complex and $q_0(t)$ the scalar parts of $q(t)$. The unit quaternion should satisfy:

$$q_0(t)^2 + \|\bar{q}(t)\| = 1$$

Finally, the quadrotors dynamic model is:

$$\ddot{p} = q \otimes \frac{F}{m} \otimes q^* + g$$

$$\dot{q} = \frac{1}{2} q \otimes \omega$$

$$\omega_b = J^{-1}(\tau - \omega_b \ast J\omega_b)$$

The quaternion derivative is derived based on the angular velocity $\omega(t)$. The designed control inputs are given by Equation (13) with $k_T$ as the thrust constant, $k_D$ the the drag constant, and $l$ the distance from the motor axis to the quadrotors center of mass:

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} k_T(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\ lk_T(\omega_2^2 - \omega_3^2 - \omega_4^2) \\ lk_T(\omega_3^2 - \omega_2^2 - \omega_4^2) \\ k_D(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2) \end{bmatrix}$$

Based on Lyapunov stability analysis, the position and velocity control laws can be derived as indicated in [26] as follows:

$$v_d = c_1 p e_1 + \dot{p}_d$$
\[
\dot{p} = e_1(I - c_1^2) + e_2(c_1P + c_2P) + \dot{p}_d
\] (15)

where \(e_1 = p_d - p\) is the position error, \(e_2 = v_d - v\) is the velocity error, and \(c_1P, c_2P\) are positive constant coefficients.

### 4.2. Formation Control

After following the leader for the same or different altitude, the formation controller’s objective is to maintain the X-Y topology. This is accomplished by maintaining a specific distance \(d\) and angle \(\alpha\) between the leader and each follower. Thus:

\[
d_x = -(X_L - X_F) \cos \psi_L - (Y_L - Y_F) \sin \psi_L
\]

\[
d_y = (X_L - X_F) \sin \psi_L - (Y_L - Y_F) \cos \psi_L
\] (16)

where the X and Y coordinates of the desired distance are \(d_x\) and \(d_y\), respectively. For the purposes of this study, an SMC controller is used to maintain formation topology in the face of external disturbances and environmental uncertainties. The \(x\) and \(y\) control errors should satisfy the following constraints:

\[
\lim_{t \to \infty} \| e_x \| = \| d_x^d - d_x \| = 0
\]

\[
\lim_{t \to \infty} \| e_y \| = \| d_y^d - d_y \| = 0
\] (17)

where \(d_x^d\) and \(d_y^d\) are the desired distance in the X and Y coordinates, respectively.

For this purpose, first-order SMC is introduced. The time varying surface \(S(t)\) is selected by

\[
s(e, t) = (\frac{d}{dy} + \lambda)^{n-1} e
\] (18)

Then, the derivative function is given by:

\[
\dot{s} = \dot{e} + \lambda e
\]

\[
\frac{1}{2} \frac{d}{dy} s^2 \leq -\eta \| s \|
\] (19)

Finally, the formation topology is controlled for each agent as follows:

\[
\dot{X}_{Fi} = \dot{X}_L + \lambda_x (\ddot{X}_L - \dot{X}_{Fi})
\]

\[
\dot{Y}_{Fi} = \dot{Y}_L + \lambda_y (\ddot{Y}_L - \dot{Y}_{Fi})
\] (20)

Once the position control problem is solved, a transformation to attitude control can be obtained. This means a minimum sharing data between agents:

\[
\dot{\theta}_{Fi} = \dot{\theta}_L + \lambda_{\theta} (\ddot{\theta}_L - \dot{\theta}_{Fi})
\]

\[
\dot{\phi}_{Fi} = \dot{\phi}_L + \lambda_{\phi} (\ddot{\phi}_L - \dot{\phi}_{Fi})
\] (21)

with \(\lambda_{\theta} > 0\) and \(\lambda_{\phi} > 0\) being selective gains for the formation control.

### 4.3. Formation Transformation

This section describes the scenario when one of the swarm agents unexpectedly fails. In such situations, the agent performs an emergency landing to avert crashes, and the other agents must switch their topologies. Additionally, when the failed agent is a leader, a leader election procedure is required. This is carried out as outlined in Algorithm 1.
Algorithm 1 Leader Election

N: Number of agents, A: list of N agents where each one has an ID from 1 to N, T = \{T(N), T(N−1), ..., T(1)\}: list of N topology according to the agent’s number, P(T): list of N paths according to each topology.

1: idL = 1
2: L = A_{idL}
3: Y = f(Batt, Health, Mission)
4: while true do
5: ex = T\(N\)
6: ex = P(ex)
7: L = [T_{ex}, T, P_{ex}, P, A, idL]
8: ALReady = 0
9: for i ← 1 N 1 do
10: if i ≠ idL then
11: send(i, ready = false) by L
12: wait(i, ready(i)) = true by L
13: ALReady = ALReady + 1
14: end if
15: end for
16: if ALReady = N − 1 then
17: for i ← 1 N 1 do
18: if i ≠ idL then
19: send(P_{ex}(i), i) by L
20: end if
21: end for
22: if Failure = idL then
23: IdY = 0
24: for j ← 1 N 1 do
25: if i ≠ idL then
26: State\_i = election
27: idY = max(Y(A\_i), IdY)
28: end if
29: end for
30: L = A_{idY}
31: idL = idY
32: end if
33: A.removeElement(A_{Failure})
34: N = N − 1
35: end if
36: end while

First of all, a set of N UAVs are identified from 1 to N. The desired topology \(T\) depends on the UAVs number. Each topology has its related paths \(P(T)\). The head of the UAV list \(A\_1\) is considered a leader \(L\). The leader is supposed to have the knowledge about: the UAVs IDs, the desired topology \(T_{ex}\), path \(P_{ex}\), and all the possible paths and topologies \(P\) and \(T\). The performance function \(Y\) depends on the battery remaining time, the UAV’s health (i.e., no failure presented), and the percentage of mission executing.

The leader sends a ready message to all UAVs before the mission to determine whether or not they are prepared (line 9). Once all UAVs are operational (line 15), the leader follows the reference path and maintains formation topology (line 18).

A test loop is required during the mission to test the state of each drone. If one of the UAVs fails (line 21), an emergency landing occurs to avoid a crash. Depending on the failing agent, one of two scenarios can occur. If the failing agent is a leader, the election procedure begins (line 25), and the agent with the best performance \(Y\) is chosen as the leader (line 26–31). If the failing agent is a follower, it will be deleted from the list of UAVs (line 32). Depending on the number of remaining UAVs, the intended topology will be switched in both scenarios (return to line 3).
5. Simulation Results

The simulation results for quadrotor creation using Matlab/Simulink are shown in this section. Many scenarios have been run based on formation control and the many limitations that can occur during a mission. All of the following scenarios make use of four quadrotor UAVs. The communication lines between all of the UAVs are expected to be secure. The goal is for all of the UAVs to be able to switch formations at any time. A summary of the simulated cases is provided below:

1. **Scenario 1:** In this scenario, UAVs are deployed as fixed-position stationary nodes that serve as communication relays. Four UAVs under rectangular topology take off from various points and fly to a predetermined altitude. The goal is to keep the optimal fixed position for maximum network connectivity.

2. **Scenario 2:** The mission to be executed during this scenario is that every UAV acts as an aerial MEC server. Nonetheless, one of the swarm agents has an unanticipated engine failure. The remaining UAVs must address this situation. Leader selection and topology switching are implemented.

3. **Scenario 3:** The third scenario simulates the UAVs as mobile nodes in a Cellular-Connected MEC. Each UAV is designed to follow a desired path with different hovering positions to serve for IoT devices. The Swarm must travel to its intended destination while avoiding external obstacles and agents collisions.

5.1. **Scenario 1: Relayed MEC**

This scenario aims to test the ability of the designed framework to deploy the UAVs in a Relayed MEC. For this case, no leader is designed, and all the UAVs have knowledge about the desired path to follow. The UAVs from 1 to 4 start from the initial positions

\[
F_1(0) = [10; -10; 0]^T, \quad F_2(0) = [-10; -10; 0]^T, \quad F_3(0) = [-10; 10; 0]^T \quad \text{and} \quad F_4(0) = [10; -10; 0]^T,
\]

respectively, and meet each other at an altitude \(z = 10\) m, while keeping a rectangular separation of 10 m. An extra edge controller is added to assure the convergence over the meeting points.

Figure 6 illustrates the success of UAVs in relaying MUs to the MEC server, as all quadrotors were able to reach the intended offloading spots and cover the entire proposed region. The formation errors between UAVs 1–2 and 3–4 converge to zero in less than six seconds. Due to the synchronization of the altitude control for all swarm agents, the error in altitude is kept at zero throughout the course of the mission. Such position hold accuracy, while preserving separation distances, would assist MUs in offloading their computationally intensive tasks and preventing connection overlap.

![Figure 6. Relayed MEC under distributed formation.](image-url)
5.2. Scenario 2: Assisted MEC

In this scenario, the swarm is supposed to start from the following positions: 
\[ F_1(0) = [50; 0; 0]^T, F_2(0) = [40; -10; 0]^T, F_3(0) = [60; -10; 0]^T, \] 
\[ F_4(0) = [50; -20; 0]^T. \]

Every UAV is denoted as an MEC server to mobile users among its trajectory. For the mission, the leader is to achieve the desired point \( P_f(x_0, y_0, z_0) = [50; 100; 10]^T \). The swarm is supposed to hold the diamond topology with a separation distance of 10 m between the agents. At \( t = 25 \) s, an engine failure occurs to the leader. According to Algorithm 1, the leader lands to avoid the crash, a new leader is then elected, and the formation is switched from diamond to triangular. Depending on the elected leader, three possible cases can be carried out:

5.2.1. Case 1

Let us suppose that Follower 1 is elected as a new leader. As depicted in Figure 7, the swarm was able to switch its topology from diamond to triangular immediately following the leader’s emergency landing. Figure 8 demonstrates that the position transfer between the new leader and Follower 2 took only 5 s, but the separation distance between the new leader and Follower 3 (10 m) was strictly observed during the switching process. After \( t = 30 \) s, the new triangular topology was totally established.

Figure 7. Leader election Case 1.

Figure 8. Leader election Case 1—tracking errors.
5.2.2. Case 2

In this case, Follower 2 is supposed to be elected as a new leader. Figures 9 and 10 illustrate the simulation case. The obtained results are similar to those obtained in the first case. The triangular topology was achieved in $t = 30\, s$, and the position's switching between the new leader and Follower 3 was made in 5 s. The separation distance between the new leader and Follower 1 was also maintained during the switching operation (Figure 10).

![Figure 9](image)

**Figure 9.** Leader election Case 2.

![Figure 10](image)

**Figure 10.** Leader election Case 2—tracking errors.

5.2.3. Case 3

For this last case, Follower 3 is supposed to be elected as the new leader. As shown in Figure 11, Follower 3 is positioned behind the leader, thus, there is no need to switch positions. Figure 12 demonstrates that the separation distance was maintained with Follower 3 and Follower 1 with high accuracy.
5.3. Scenario 3: Cellular-Connected MEC

The UAVs in this scenario are considered as mobile nodes in a cellular-connected MEC server. The four UAVs are supposed to start from the following positions: \( F_1(0) = [50; 65]^T \), \( F_2(0) = [60; 75]^T \), \( F_3(0) = [50; 85]^T \), and \( F_4(0) = [40; 75]^T \). The leader’s objective is to get to the target spot: \( P_f(x_0, y_0) = [50; 15]^T \) and avoid the obstacle (\( R = 10 \text{ m} \)) situated at \( O_1(x_0, y_0) = [50; 50]^T \). The agents are separated by 10 m. The diamond configuration is expected to be held by the swarm. Otherwise, it is possible to split the topology, avoid the obstacles, and go back to the initial formation.

In order to conserve energy, the swarm was able to circumvent the circular barrier by forming a distributed configuration (Figure 13). The swarm was split into two teams, each with two leaders: one created by the original leader and follower 1, and the other by follower 2 (new leader) and follower 3 (Figure 14). The formation error of teams 1 and 2 is shown below. It can be seen that the separation distance (10 m) between the two UAVs was respected with excellent precision in both the x- and y-directions. This demonstrates the formation controller’s excellent performance.
Table 1 summarizes the results of the previous scenarios. As previously demonstrated, the UAV swarm proved capable of enabling diverse MEC topologies and accommodating all possible constraints. While tracking the created path with the least possible error/delay, the convergence time was very reasonable. The architecture is also ideal for long-range missions due to the energy consumption strategy.

Table 1. Performance results.

| MEC Architecture                  | Relayed | Assisted | Cellular-Connected |
|-----------------------------------|---------|----------|-------------------|
| Convergence time (s)              | 10      | 50       | 120               |
| Traveled distance (m)             | 10      | 70       | 80                |
| Total energy consumption (%)      | 5       | 15       | 20                |
5.4. Comparative Study

To show the usefulness of the proposed framework, a comparison with state-of-the-art methods was conducted. The comparison of the architectures’ key features is presented in Table 2, from which we can deduce that our suggested framework constitutes a comprehensive effort in comparison to other current works. First, the distributed architecture solves the shortcomings of the centralized architecture since leader and follower interaction is taken into account. In addition, the algorithms utilized for path planning and formation control are user-friendly, suitable for a large number of robots, and can be implemented within an embedded system. Our framework is the only one that has strengthened safety precautions by including a leader election process.

Table 2. Article architecture comparison.

| Author           | Wu et al. (2020) | Wen et al. (2019) | Tran et al. (2021) | Proposed Framework |
|------------------|------------------|-------------------|-------------------|-------------------|
| Vehicle Type     | UAV              | UGV               | UAV/UGV           | UAV               |
| Number           | 8                | 4                 | 3                 | 4                 |
| Path planning    | PSO              | APF               | NI                | CO                |
| Formation control | Strategy        |                   |                   |                   |
|                  | Position consensus | Position consensus | Velocity consensus | Attitude consensus |
|                  | MPC              | Robuste $H\infty$ | NI                | SMC               |
| Safety precautions | N/C             | Switching         | Switching         | Switching/Leader election |

In addition, the performance of formation control results may be found in Table 3. We can argue that the employment of attitude-based control based on SMC controller has significantly enhanced the L-F configuration’s precision, speed, and stability.

Table 3. Formation control performance comparison.

| Author           | Wu et al. (2020) | Wen et al. (2019) | Tran et al. (2021) | Proposed Framework |
|------------------|------------------|-------------------|-------------------|-------------------|
| Rise time (s)    | 5                | 3                 | 5                 | 4                 |
| Over shoot %     | 0                | 0                 | 5                 | 0                 |
| Setting time (s) | 10               | 5                 | 10                | 5                 |
| Switching time (s) | N/C           | 4                 | 5                 | 1                 |
| Tracking error (m) | x               | 0.3               | 0.1               | 0.05              |
|                   | y               | N/A               | 0.3               | 0.1               |

6. Conclusions

The issue of multi-UAV-enabled MEC architectures in a distributed formation was investigated in this study. In the case of centralized UAV formation, the leader is the sole agent who is fully aware of the mission’s details. As a result, every probable failure has an impact on mission execution. To overcome this issue, a new framework of quadrotors with a distributed L-F configuration was presented. The framework is divided into several stages, each of which takes into account the relationship between the leader and the followers. Throughout the formation control stage, a consensus-based attitude control was used to maintain the formation topology with minimal data sharing. An SMC controller was used to monitor the optimal generated path with the minimum possible error/delay. During the trajectory generation phase, a collaborative obstacle avoidance technique was developed for safety and to accommodate the multiple environmental constraints. The swarm’s agents were able to switch topology, avoid obstacles, and return to the desired formation without colliding. Furthermore, in response to an agent’s unexpected death, the agents were able
to elect new leadership. The leader election phase was designed to indicate topological changes caused by external obstacles. In comparison to the vast majority of comparable suggested works, all produced results were deemed satisfactory. Although this study focused on the use of UAVs under various MEC architectures, the technical computing approach can be generalized and applied to many other IoT specifications and access control strategies, such as networking and data gathering approaches, educational platforms, healthcare systems, transportation services, and many other real-world applications. Therefore, incorporating the proposed architecture into an MEC-based real-world application would be an intriguing endeavor.

Author Contributions: Conceptualization, K.C.; data curation, K.C. and S.F.; methodology, S.M.; software, K.C.; supervision, M.L.; validation, K.C., M.L., S.M., and S.F.; writing—original draft, K.C.; writing—review and editing, S.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by Princess Nourah Bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R196), Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No data available.

Acknowledgments: The authors would like to acknowledge the Princess Nourah Bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R196), Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia.

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

References

1. Abrar, M.; Ajmal, U.; Almohaimeed, Z.M.; Gui, X.; Akram, R.; Masroor, R. Energy efficient UAV-enabled mobile edge computing for IoT devices: A review. IEEE Access 2021, 9, 127779–127798. [CrossRef]
2. Faraci, G.; Grasso, C.; Schembra, G. Fog in the clouds: UAVs to provide edge computing to IoT devices. ACM Trans. Internet Technol. TOIT 2020, 20, 1–26. [CrossRef]
3. Kim, S.; Oh, H.; Suk, J.; Tsourdos, A. Coordinated trajectory planning for efficient communication relay using multiple UAVs. Control. Eng. Pract. 2014, 29, 42–49. [CrossRef]
4. Rahbari, D.; Mahtab Alam, M.; Le Moulliec, Y.; Jenihhin, M. Edge-to-Fog Collaborative Computing in a Swarm of Drones. In International Conference on Model and Data Engineering; Springer: Berlin/Heidelberg, Germany, 2021; pp. 78–87.
5. Wang, Y.; Fang, W.; Ding, Y.; Xiong, N. Computation offloading optimization for UAV-assisted mobile edge computing: A deep deterministic policy gradient approach. Wirel. Netw. 2021, 27, 2991–3006. [CrossRef]
6. Chang, H.; Chen, Y.; Zhang, B.; Doerrmann, D. Multi-UAV mobile edge computing and path planning platform based on reinforcement learning. IEEE Trans. Emerg. Top. Comput. Intell. 2021, 6, 489–498. [CrossRef]
7. Guo, H.; Liu, J. UAV-enhanced intelligent offloading for Internet of Things at the edge. IEEE Trans. Ind. Inform. 2019, 16, 2737–2746. [CrossRef]
8. Peng, C.; Huang, X.; Wu, Y.; Kang, J. Constrained Multi-Objective Optimization for UAV-Enabled Mobile Edge Computing: Offloading Optimization and Path Planning. IEEE Wirel. Commun. Lett. 2022, 11, 861–865. [CrossRef]
9. Tropea, M.; De Rango, F.; Nevigato, N.; Bitonti, L.; Pupo, F. Scare: A novel switching and collision avoidance process for connected vehicles using virtualization and edge computing paradigm. Sensors 2021, 21, 3638. [CrossRef]
10. Gong, C.; Wei, L.; Gong, D.; Li, T.; Feng, F. Energy-Efficient Task Migration and Path Planning in UAV-Enabled Mobile Edge Computing System. Complexity 2022, 2022. [CrossRef]
11. Song, K.T.; Sun, Y.X. Coordinating multiple mobile robots for obstacle avoidance using cloud computing. Asian J. Control. 2021, 23, 1225–1236. [CrossRef]
12. Miao, Y.; Hwang, K.; Wu, D.; Hao, Y.; Chen, M. Drone Swarm Path Planning for Mobile Edge Computing in Industrial Internet of Things. IEEE Trans. Ind. Inform. 2022, 1–11. doi: 10.1109/TII.2022.3196392. [CrossRef]
13. Jiang, C.; Fang, Y.; Zhao, P.; Panneerselvam, J. Intelligent uav identity authentication and safety supervision based on behavior modeling and prediction. IEEE Trans. Ind. Inform. 2020, 16, 6652–6662. [CrossRef]
14. Jiang, Y.; Wang, S.; Lei, L. A Multi-UAV Formation Maintaining Method Based on Formation Reference Point. J. Phys. Conf. Ser. 2020, 1621, 012014. [CrossRef]
15. He, Y.; Liu, J.; Tong, E.; Niu, W.; Huang, X.; Zhou, Y.; Li, C.; Chang, L. Adapt Swarm Path Planning for UAV Based on Artificial Potential Field with Birds Intelligence Extensions. In International Symposium on Security and Privacy in Social Networks and Big Data; Springer: Berlin/Heidelberg, Germany, 2020; pp. 340–354.

16. Liu, H.; Lu, Y.; Zhao, W. Robust visual servoing formation tracking control for quadrotor UAV team. *Aerosp. Sci. Technol.* **2020**, *106*, 106061. [CrossRef]

17. Parrany, A.M.; Alasty, A. Decentralized aggregation and leader-following control of a swarm of quadcopters with nonlinear under-actuated dynamics. *Aerosp. Sci. Technol.* **2020**, *107*, 106317. [CrossRef]

18. Ai, X.; Yu, J. Flatness-based finite-time leader–follower formation control of multiple quadrotors with external disturbances. *Aerosp. Sci. Technol.* **2019**, *92*, 20–33. [CrossRef]

19. Brust, M.R.; Strimbu, B.M. A networked swarm model for UAV deployment in the assessment of forest environments. In Proceedings of the 2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Singapore, 7–9 April 2015; pp. 1–6.

20. Shames, I.; Teixeira, A.; Sandberg, H.; Johansson, K.H. Distributed leader selection without direct inter-agent communication. In Proceedings of the 2nd IFAC Workshop on Distributed Estimation and Control in Networked Systems, NecSys’10, Annecy, France, 13–14 September 2010; pp. 221–226.

21. Franchi, A.; Giordano, P.R. Online leader selection for improved collective tracking and formation maintenance. *IEEE Trans. Control. Netw. Syst.* **2018**, *5*, 3–13. [CrossRef]

22. Choutri, K.; Lagha, M.; Dala, L.; Lipatov, M. Quadrotors UAVs Swarming Control Under Leader-Followers Formation. In Proceedings of the 2018 22nd International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania, 10–12 October 2018; pp. 794–799.

23. Xu, M.; Fang, Y.; Li, J.; Zhao, X. Chattering Free Distributed Consensus Control for Attitude Tracking of Spacecraft Formation System with Unmeasurable Angular Velocity. *Int. J. Control. Autom. Syst.* **2020**, *18*, 2277–2288. [CrossRef]

24. Hou, Z. Modeling and formation controller design for multi-quadrotor systems with leader-follower configuration. Ph.D. Thesis, Université de Technologie de Compiègne, Compiègne, France, 2016.

25. Stojkovi, I. Formation Control of Robotized Aerial Vehicles Based on Consensus-Based Algorithms. *FME Trans.* **2017**, *45*, 559–564. [CrossRef]

26. Choutri, K.; Lagha, M.; Dala, L. Multi-layered optimal navigation system for quadrotor UAV. *Aircr. Eng. Aerosp. Technol.* **2019**, *92*, 145–155. [CrossRef]

27. Pham, H.; Smolka, S.A.; Stoller, S.D.; Phan, D.; Yang, J. A survey on unmanned aerial vehicle collision avoidance systems. *arXiv Preprint 2015*, arXiv:1508.07723.

28. Lao, M.; Tang, J. Cooperative Multi-UAV Collision Avoidance Based on Distributed Dynamic Optimization and Causal Analysis. *Appl. Sci.* **2017**, *7*, 83. [CrossRef]

29. Misir, O.; Gökrem, L. Flocking-Based Self-Organized Aggregation Behavior Method for Swarm Robotics. *Iran. J. Sci. Technol. Trans. Electr. Eng.* **2021**, *45*, 1427–1444. [CrossRef]

30. Dang, A.D.; La, H.M.; Nguyen, T.; Horn, J. Distributed Formation Control for Autonomous Robots in Dynamic Environments. *CoRR 2017*, arXiv:1705.02017.

31. Singh, Y.; Bibuli, M.; Zereik, E.; Sharma, S.; Khan, A.; Sutton, R. A novel double layered hybrid multi-robot framework for guidance and navigation of unmanned surface vehicles in a practical maritime environment. *J. Mar. Sci. Eng.* **2020**, *8*, 624. [CrossRef]

32. Di Lillo, P.; Pierrì, F.; Antonelli, G.; Caccavale, F.; Ollero, A. A framework for set-based kinematic control of multi-robot systems. *Control. Eng. Pract.* **2021**, *106*, 104669. [CrossRef]

33. Liu, T.; Han, D.; Lin, Y.; Liu, K. Distributed multi-UAV trajectory optimization over directed networks. *J. Franklin. Inst.* **2021**, *358*, 5470–5487. [CrossRef]

34. de Freitas, E.P.; Basso, M.; da Silva, A.A.S.; Vizzotto, M.R.; Corrêa, M.S.C. A Distributed Task Allocation Protocol for Cooperative Multi-UAV Search and Rescue Systems. In Proceedings of the 2021 International Conference on Unmanned Aircraft Systems (ICUAS), Athens, Greece, 15–18 June 2021; pp. 909–917.