Learning-Based Formation Control of UAV-fleet

Abdulhakeem Abdulazeez*, Nicola Roberto Zema*, Tara Ali-Yahiya*, Steven Martin*
*Université Paris-Saclay, CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique, 91190 Gif-sur-Yvette, France
{abdulhakeem.abdulazeez, zema, tara.yahiya, steven.martin }@lisn.fr

***Abstract*—In the context of wireless networked robotics, complex missions, such as autonomous Search and Rescue, may require the use of multiple Unmanned Aerial Vehicles (UAVs) to achieve higher efficiency and Quality of Service (QoS) while assuring mission-specific imperatives like the maximization of area covered in a single passage of the fleet over area subsections. In this paper, we present a learning-based formation control protocol that adapts the principle of Q-learning to pilot an autonomous fleet of networked UAVs to maintain formation throughout a mission where large quantities of data need to be exchanged. Also, the protocol tries to ensure rotational formation control by leveraging only the signal strength extrapolated from the UAV communications. A leader-follower model is used to control the fleet. One UAV serves as the leader, and the remaining as the followers. The followers use the Received Signal Strength Indicator (RSSI) values obtained from their neighbors to autonomously determine the leader's direction of movement and maintain formation orientation to avoid area coverage overlapping. We carried out several simulation experiments to evaluate the performance of the proposed scheme in terms of QoS and convergence time of the formation under varying velocities.**

***Index Terms*—Wireless Networked Robots, UAV, formation control, Q-learning, fleet control.**

I. Introduction

The advances in robotics and aeronautics technologies have led to an increase in the application of *Unmanned Aerial Vehicles* (UAVs), also known as drones or quadcopters, for more and more complex civilian operations or tasks such as surveillance, logistics, media, sports, and *Search and Rescue* (SAR) operations. Most of these are carried out in large areas and require the exchange of images or video data in real-time for instant decision-making. Tasks like SAR are even more time critical because they involve reaching and operating over a potentially life-threatening target. It has been speculated and researched that using a single drone will not be as effective for these operations due to the limited size and capability of single UAVs [1]. More so, a single *Unmanned Aerial Vehicle* (UAV) mission requires a powerful drone that needs to fly to a very high altitude, carrying a very high-resolution camera [2]; also, in the event of failure, the whole operation halts. These reasons led to using multiple UAVs (multi-UAVs) for such complex operations.

Multi-UAV operations require controlling a fleet of UAVs to maintain a predefined formation pattern throughout the mission or adapt to changes in the mission or the environment [3]. Several frameworks and solutions are proposed in the literature, focusing on the different challenges facing the design of multi-UAV operations. These include target identification [4] [5], agent autonomy [6], localization and formation control [7] [8]. Localization and formation control of a fleet of UAVs has recently attracted more research efforts because of its importance to any design or solution involving multiple autonomous vehicles. Localization of UAVs nominally requires absolute positioning techniques such as *Global Positioning System* (GPS), although GPS is generally considered ineffective indoors or in places with physical obstructions [9] [10]. Other techniques like *Angle-of-Arrival* (AoA) and *Time-of-Arrival* (ToA), which provide location or direction information relative to a reference frame leveraging radio signals, but they require additional hardware such as *Ultra-wideband* (UWB) devices [10]. Therefore, location fingerprinting techniques that are more cost-effective and indoor-friendly were proposed [11]. The location fingerprinting methods leverage the *Received Signal Strength Indicator* (RSSI) values that each robot computes during communications.

RSSI-based localization and formation control techniques provide promising results, especially for outdoor applications and when considering hardware cost and complexity. For this reason, several proposals have been made in the literature. For example, The work in [3] proposed a consensus-based UAV formation and obstacle avoidance algorithm. The algorithm addresses the constraints on flight velocity, acceleration, and *angular rate* by adjusting their values after running a consensus algorithm while avoiding collisions between UAVs. It decomposes the UAV's motion into a multidimensional axis and designs control laws for each direction to account for the state and position gaps between UAVs. The consensus algorithm is combined with *Particle Swarm Optimization* (PSO) to avoid static obstacles. It is also combined with *Model Predictive Control* (MPC) to avoid dynamic ones. The combined algorithms allow UAVs to maintain the desired formation while avoiding obstacles. The authors in [7] proposed a centralized multi-UAV formation control scheme. The scheme adopts a leader-follower model with three followers positioned at the vertices of an equilateral triangle and a virtual leader at the center. The UAV formation passes through three phases to form and keep the desired shape. The first is the loss phase; in this phase, each UAV in the fleet takes off from different points. The UAVs then switch to the assembly phase, establishing a wireless ad-hoc connection. The virtual leader then receives the flight trajectory information from the *Ground Control Station* (GCS) and broadcasts it to followers. The followers follow the leader to form the expected formation shape. Finally, the keeping phase ensures that the fleet remains in formation for the remaining period of the operation. The scheme employed a

978-3-903176-59-1

control protocol based on a back-stepping approach to reduce the relative position error caused by environmental effects. A GPS based multi-UAV trajectory recovery scheme is proposed in [12]. The scheme aims to control UAVs that are out of the transmission range of the GCS. The UAVs are GPS enabled and are connected to one another in an ad-hoc mode, forming a linear pattern to cover more operation area. The fleet is connected to the GCS through one of the UAVs that forms a direct connection with GCS. Each UAV periodically collects and transmits GPS data through its neighbors down to the GCS. The GCS processes the received data, determines the UAV's position, and sends the information back to the sender UAV. The UAVs along the path only act as relays. Each UAV also maintains a neighbor table containing the location information of its neighbors. The table is updated periodically, and an existing record is deleted when its *Time-To-Live* (TTL) expires. Therefore, when the last record in the neighbor table expires, the UAV considers itself out of formation and then triggers the self-recovery protocol. To rejoin the formation, the out-of-range UAVs moves to the location of the last UAV in its table. If it receives new neighbor information, it rebuilds its table to maintain form; else, it returns to base. The scheme achieves extended operational coverage and a low-cost recovery. Authors in [13] proposed a geometrical localization scheme based on the RSSI for a fleet of UAVs. The scheme separates the UAVs into two groups and controls the UAVs in each group to maintain a circular trajectory using the differences in their RSSI values. In [14], a localization scheme for fixed-wing UAVs using RSSI is proposed. Unlike previous methods that rely on detailed knowledge of transmitted power, antenna, and channel characteristics, the proposed scheme compares received powers by antennas on a linear array. Then, it uses a control algorithm to ensure consensus on received power, which allows for determining the device's location without extensive parameter information. The work in [15] explores the use of intelligent robot swarms for search and rescue missions. The scheme uses a genetic algorithm to autonomously control the UAV fleet using the RSSI values received by each UAV from its immediate neighbors. However, because each UAV relies only on its immediate neighbor to position itself, if wrong RSSI value is received from any UAV, all UAVs that connect to the formation through it will break out of formation. The scheme proposed in [16] utilizes reinforcement learning to enable a UAV to determine its trajectory autonomously. The study incorporates UAV-to-ground channel characteristics, empirical path loss, shadowing models, and select waypoints that minimize the average location errors.

In [8], a behavior-based formation control scheme that employs a Q-learning strategy to control a fleet of UAVs is proposed. The scheme's learning strategy uses only the RSSI values being exchanged periodically among the connected UAVs to compute the distance between the current UAV's position and its expected position. During the learning phase, a vehicle travels in different possible paths (N, E, W, and S) and obtains its respective distances to its expected location. The UAV then travels along the path with the shortest distance. The vehicle repeats this process until it reaches its expected position in the formation. The RSSI values the proposed scheme uses make it less complex and cost-effective than other schemes that use multiple parameters. However, the scheme may cause a longer convergence time or converge into a correct formation pattern but in a wrong orientation, which will consequently cause an increase in exploration because its failure considers formation orientation. Therefore, we propose a learning-based formation control protocol that employs an orientation guide mechanism that forces the follower drones into proper formation and orientation to the direction of flight. The contributions of this paper are three-pronged and stated as follows:

1) A Q-Learning-based speed-control mechanism trains the follower UAVs to maintain the same speed as the leader autonomously.
2) A Q-learning algorithm for direction detection and formation adjustment.
3) A mechanism that monitors a follower UAV to determine its status and then decides to trigger either of the Q-learning algorithms in phases one and two.

The rest of this paper is organized as follows. In Section II, we present a review of related literature on the formation control of UAV-fleet. Then, we discuss the system model and define the research problem in Section III. Our proposed protocol is illustrated in Section IV. The simulation setup is presented in Section IV. In Section VI, we present and discuss the simulation results and then conclude the paper in Section VII.

II. System Model

We consider a UAV fleet consisting of one leader and n followers communicating using IEEE 802.11-family in ad-hoc mode deployed for a SAR mission. In the simplest case, the UAVs are configured to start from a predefined formation shape and are expected to maintain it throughout the mission. The number of followers, the shape, and the distance between the UAVs are determined a priori based on the search area's size, the shape, and the mission's expected duration. We assume that the position of the search target is unknown, and it is crucial to find it in the shortest possible time. In our system model, the GCS controls the leader directly to follow a parallel-sweep search pattern, which is very effective for this kind of mission [17]. At the same time, the followers are expected to autonomously follow the leader to maintain the fleet's initial formation shape. For clarity, see Figure 1. The figure shows a fleet of three UAVs deployed for SAR mission.

The fleet is expected to keep a triangular formation at an altitude of δ meters. The leader UAV occupies the top vertex. In contrast, followers occupy the bottom vertices of an isosceles triangle. For the rest of the documents, we will refer to the triangular formation for the sake of simplicity.

For SAR operations, the UAVs are supposed to be equipped with downward-looking cameras, and the lengths of the triangle are defined such that the sum of all horizontal fields of view

(FVs) of the three UAVs equals the width of the search area Xm (see Eq. 1).

$$Xm = Fv_{h1} + Fv_{hL} + Fv_{h2} \tag{1}$$

The Fv is considered as the area directly below the UAV covered by a device camera [4]. It is measured along the vertical and horizontal axis of the covered areas based on the vertical and horizontal angle of views, respectively. We used only the horizontal field of view (Fv_h) of each UAV camera here because it defines the width of the area covered along the direction of flight. Therefore, Fv_hL is the leaders' horizontal field of view, while Fv_{h1} and Fv_{h2} denote the horizontal field of view of followers 1 and 2, respectively. Fv_h is defined as in Eq. 2:

$$Fv_h = 2\delta tan(\frac{\theta_h}{2}) \tag{2}$$

where δ is the UAV altitude, θ_h is the horizontal angle of view of the camera, defined as Eq. 3:

$$\theta_h = 2tan^{-1}(\frac{\alpha_h}{2\beta}) \tag{3}$$

where α_h is the width of the camera lens and β is the focal length of the camera.

From Figure 1, it is easy to observe that if the sum of the horizontal field of views for follower 1, leader, and follower 2 equals the width of the search area X_m and the UAVs are with the right orientation, it is possible to search the whole search with the minimum number of passages. However, it is necessary that the UAVs keep a formation oriented with respect to the direction of flight of the leader.

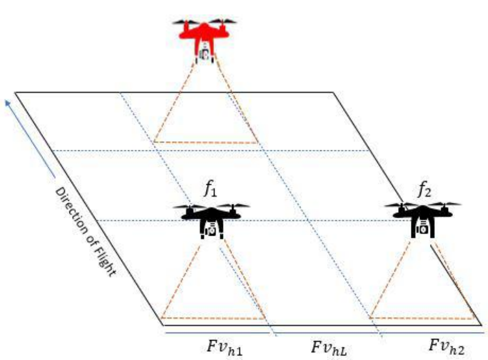

Fig. 1. Fleet of three UAVs Deployed for SAR

A. Problem Definition

As stated in the previous sections, our work is based on [8]. In that work, a Q-learning strategy was proposed to control a fleet of UAVs using only the RSSI values obtained through the IEEE 802.11 communication within the fleet. During the scheme's learning phase, a follower UAV travels in different possible paths (N, E, W, and S) and obtains its respective distances to its expected location by computing the Euclidean distance between the set of its received RSSI vales at its current position and that of its expected position. The UAV then travels along the path with the shortest distance. It repeats this process until it reaches its expected position in the formation. The scheme is light and cost-effective as it avoids using external positioning or localization devices. However, the scheme did not consider the formation orientation, which could cause the fleet to move in the correct shape but in the wrong orientation, as shown in Figure 2.

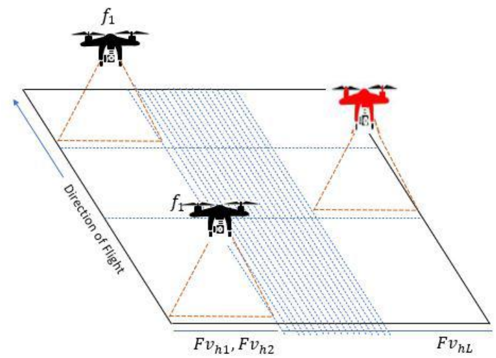

Fig. 2. Fleet of three UAVs in a *wrong* Orientation

In contrast to the formation depicted in Figure 1, Figure 2 shows the fleet in a wrong orientation relative to the direction of flight. The two followers explore the same section of the search area, causing the shaded region of the search area to be left unexplored, consequently leading to an increase in exploration time and cost. Also, Since the follower's position is strictly determined by its proximity to its neighbors, if a follower UAV falls out of formation, the other follower also considers itself out of formation, and this will cause an increase in the convergence time of the learning process. To address these problems, we propose a learning-based formation control protocol that employs an orientation guide mechanism that forces the follower drones into proper formation and orientation to the direction of flight.

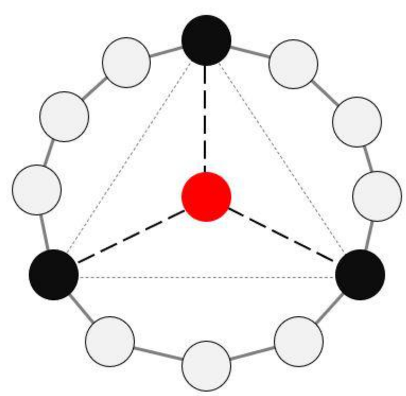

Fig. 3. Fleet of four UAVs in Equilateral Triangular Formation

III. Proposed Protocol

In our context, we adopt a four UAVs formation [7], as shown in Figure 3. Three UAVs (the followers) occupy the vertices of an equilateral triangle of predetermined length, while the leader UAV is positioned at the center of the triangle. This topology provides the followers equal proximity to the leader. The receives control information from GCS to follow the parallel-sweep search model shown in Figure **??**. Each UAV maintains a neighbor-RSSI table, which maps its neighbors' IDs and received RSSI values. The record in a UAV's neighbor-RSSI table determines its current position at any given time during the mission. The aim of our proposed protocol is to make the follower UAVs autonomously follow the leader while maintaining the initial formation configuration as it maneuvers through the search area. To achieve this, we must address two basic challenges: if the followers move in the same direction as the leader, they must maintain the same velocity as the leader to keep the formation. If the leader changes direction, the followers must detect and align themselves consistently with the direction of flight of the leader. For an improvable algorithmic simplicity, we assume that the fleet moves at a fixed altitude along eight possible directions along the yaw axis at an angular difference of 45^o, as shown in Figure 4. We incorporate two modules into our proposed protocol to address each of the challenges highlighted above: *Q-learning-based speed control algorithm* (QSCA) and *Direction Detection and Formation Alignment* (DDFA).

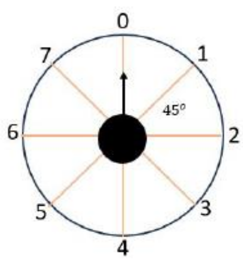

Fig. 4. Fleet Possible Direction of Flight

Each of these modules is triggered depending on the state of the formation. If the fleet is in initial formation, the QSCA is triggered by the followers to adjust its velocity of the follower adaptively. When the formation transitions into a different state due to a change in the leader's direction of flight, the DDFA is triggered to control the follower accordingly. The operation of each module is detailed in the next sections.

A. Speed Control Phase

As mentioned in the previous Section, GCS controls the leader UAV, and the followers autonomously follow the leader while maintaining their initial formation. Whenever the leader's speed changes as shown in Figure 5, the followers must keep up with the changes; otherwise, they automatically fall out of the formation shape. Figure 5(a) shows the reference state of the formation. This state is maintained if the velocity of the leader and the follower UAVs are the same. Figure 5(b) and 5(c) show the formation topology when the leader is faster and slower than the followers respectively. QSCA aims to learn the optimal policy that guides the follower UAVs on the best actions to take at any given instance of the environment to maintain the initial formation of the mission. We model the QSCA's Markovian Decision Process (*Markovian Decision Process* (MDP)) as follows:

System State: In every time step t, the system state $s_n(t)$ $\in S$ as perceived by each UAV characterizes the environment in terms of the proximity of the followers to the leader, as the velocity of the leader varies. The state Sn of the environment for ith UAV is defined as a tuple $S_{n\omega}(x_{i1}, x_{i2}, x_{i3})$ that holds RSSI values for its connection to its three neighbors.

Action: Each follower UAV chooses an action $a_n(t) \in A$ in its current state $s_n(t)$ based on policy π. The action taken by a follower in a particular state of the environment is either to maintain velocity, increase velocity by 1 m/s, increase velocity by 2 m/s, decrease velocity by 1 m/s, or decrease velocity by 2m/s.

Reward: Whenever a follower UAV takes an action a_t in a particular state s_t, it receives an immediate reward $R(s_t, a_t)$ which we defined as:

$$R(s_t, a_t) = \begin{cases} 100 & \text{If reference state} \\ -1 & \text{otherwise} \end{cases} \tag{4}$$

B. Direction Detection and Formation Alignment phase

This section discusses our Q-learning-based direction detection and formation alignment algorithm. We assume that all UAVs move in only eight possible directions along the yaw axis at an angular interval of 45^o, as illustrated in Figure 4. We denote the directions as d_r, r=$\{0, 2, 3, ..., 7\}$ and the distances between the leader UAV and f_1, f_2, and f_3, respectively as $d_{01}(t)$, $d_{02}(t)$, $d_{03}(t)$. These distances are relatively equal whenever the fleet is in the correct formation.

System State: When the leader UAV changes its direction, the state $s_n(t)$ which defines its proximity to each of the followers changes. Since the relationship between any $d_{0i}(t)$ and $d_{0j}(t)$ at $t > 0$ is either equal, less, or greater. Each follower's set of received RSSI values will also change, causing the environment to transition into a different state.

Action: Since our goal is first to determine the direction of flight of the leader UAV, our Q-learning algorithm learns the optimal policy that guides the followers to take the best action d_r and angular displacement (θ) at an interval of 45^o to change the formation orientation towards the direction of flight of the leader. The value of d^*_{0i} varies with time due to environmental effect [13], even if the geometric distance between the UAVs is fixed. Therefore, we use Cost231PropagationLossModel in NS3 to approximate the RSSI oscillation.

Reward: Each follower UAV receives an immediate reword $R(s_t, a_t)$ define in Eq. 4.

IV. Simulation

This section discusses the simulation setup, implementation, and experiments carried out to evaluate our protocol's performance. As mentioned in section IV, we adopt the formation

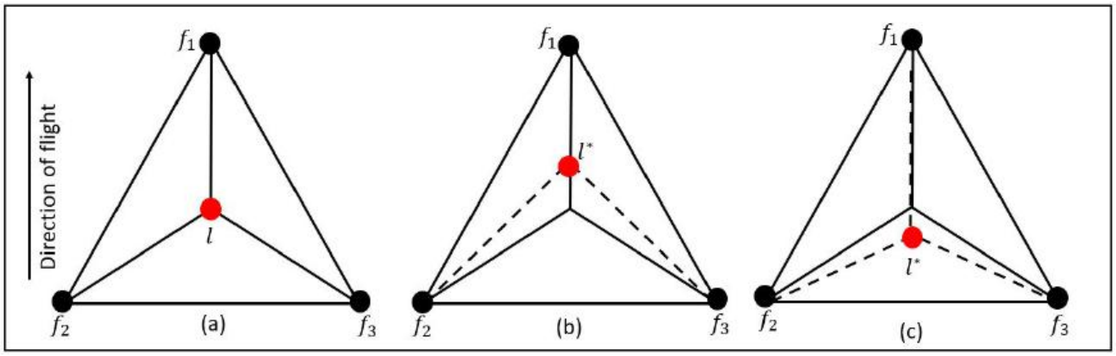

Fig. 5. Variation in the speed of the leader UAV against the followers

topology presented in [7]. The network comprises four nodes: three followers, f_1, f_2 and f_3, and a leader UAV f_0. The followers are positioned in an equilateral triangle with f_1 at the top vertex and f_1 and f_2 at the bottom. The leader is positioned at the center. We implemented this topology and our protocol in NS3 version 3.35. UAVs communicate wirelessly in an ad-hoc mode using IEEE 802.11n. This is to provide room for decentralized and flexible communication among the UAVs. Each UAV maintains a neighbor-RSSI table, which maps their neighbors' ID and RSSI values. Each UAV updates and broadcasts its neighbor-RSSI table periodically as it receives what we will onward refer to as position-info-packets from its neighbors. The records in a UAV's neighbor-RSSI table at any time of the mission determine the current state of that UAV.

Our follower flight controller continuously queries a UAV's neighbor-RSSI table to determine its current state. If the UAV is in its correct position, the QSCA is triggered to keep it up with the leader's speed. Otherwise, it triggers the DDFA to change the follower's orientation towards the direction of flight of the leader. The MDP parameters set for the training of both QSCA and DDFA are as follows:

TABLE I
MDP PARAMETERS FOR QSCA AND DDFA

Parameter	Values	
	QSCA	***DDFA***
Learning Rate(α)	0.1	0.6
Discount factor(γ)	0.9	0.9
Exploration rate(ϵ)	0.8	0.8
Number of episodes	1000	1000

For QSCA, the Q-table is initialized to zero at the beginning of the training phase. At the beginning of every episode, the follower starts with a speed of 2m/s, while the leader's start speed is randomly selected between a range of 0.5m/s and 10m/s inclusive with an interval of 0.5. The maximum leader speed is set to 2.5m/s, 5m/s, 7.5m/s, and 10m/s for different simulation scenarios. Each follower takes action based on policy π. It selects the best action only 20% of the time and randomly selects any of the five actions in the remaining 80% of the time. It receives a reward of -1 for transiting from one state to another and a reword of 100 when it reaches the goal state. The Q-value for every state is computed using

$$Q(s,a) = Q(s,a) + \alpha[r + \gamma.max[Q(s',a')] - Q(s,a)] \quad (5)$$

where: α (alpha) is the learning rate; it determines how often newly acquired knowledge replaces the previous one. γ (gamma) is the discount factor, which determines how important future rewards are compared to immediate rewards. s' is the new state reached after taking action a. a' is the action in the new state that maximizes the Q-value following the optimal policy. max[Q(s', a')] is the Q-value obtained for taking the best action in the new state s'.

The current episode terminates once it reaches the goal state, and the next episode starts until a circle of one thousand episodes is completed.

We also trained the DDFA by initializing its Q-table to zero. The Q-table was updated using Eq. 5 repeatedly for one thousand training episodes. In each episode, the leader takes any of the eight possible flight directions, as shown in Figure 4, putting the formation in a different state than the initial state (State 0). The followers detect the new direction of flight by exploring the eight possible directions while considering the variations pattern of the RSSI in their neighbor-RSSI table. It randomly selects any of the eight actions 80% of the time and takes the best action 20% of the time. For each action taken, the agent is penalized with a reward of -1. This is to encourage it to avoid unnecessary actions. It receives a reward of 100 for reaching the goal state.

It is important to note that we have chosen these values for rewards for both modules for the sake of simplicity. We plan to study the Markovian hyperparameters in future works extensively. As highlighted above, all UAVs partake in the exchange of position-info-packet, which is necessary to keep their formation. The followers are also configured to transmit video packets in real-time to the leader via Unicast. We incorporated Evalvid [18] into the NS3 simulator to simulate the video transmission for its simplicity and validations by many research works [19] that presented its effectiveness when used in NS3. We used Cost231PropagationLossModel and used the stock parameters for Evalvid.

V. Results and Discussion

In this section, we provide the simulation results to demonstrate the effectiveness of our proposed protocol in terms of formation stability and its effect on the QoS of transmitted data. We conducted a pre-simulation campaign to determine reference parameters for training our machine learning protocol. We obtained reference position parameter $P(r_1, r_2, r_3)$ for each follower UAV for different formation sizes. r_i is the estimated RSSI value a follower UAV receives from its i^{th} neighbor.

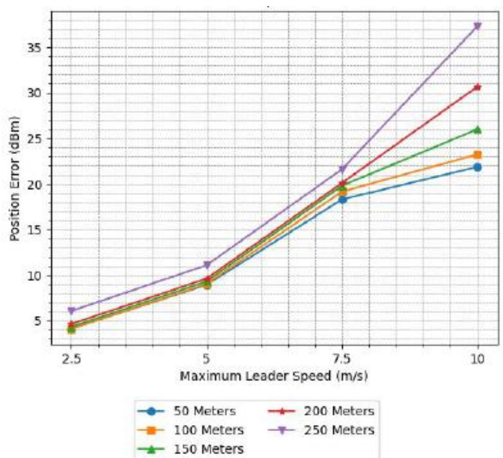

Fig. 6. Position Error at Different Speed and Formation Sizes

Figure 6 shows the position error of the formation for different leader speeds and formation sizes. For each formation size, we obtained the position error for different leader speeds to determine how much impact the leader speed has on formation stability for varying formation sizes. It can be observed that the position error obtained for leader speeds 2.5, 5, and 7,5 are similar for follower distances of 50, 100, 150, and 200 meters. The error obtained from the speeds from 7.5 to 10m/s varies. Also, for a formation size of 250 meters, there is an increase in the error for all the speeds. However, it can be observed that there is a consistent increase in the position error as the speed increases. This shows that the leader's speed negatively impacts the formation more than the formation size. We observed that for variation of 6 to 8 dBm, there is no observable difference in the formation shape. Therefore, The proposed protocol is stable at a maximum leader speed of 5m/s and for a formation size of up to 200 meters.

In a second set of simulations, we used Evalvid with stock parameters, i.e. sending the video without restrictions at the application layer. Figures 7, 8 and 9 show the end-to-end delay, jitter, and packet loss ratio incurred for transmitting video packets to the leader by the followers at varying leader speeds and formation sizes, respectively. Figure 7 reveals that both the speed and formation sizes cause an increase in delay. However, the formation size has more impact because it causes an increase in the proximity of each UAV to its neighbors. On the other hand, figure 8 shows that a relatively similar jitter was recorded from all the speed and formation sizes. The maximum

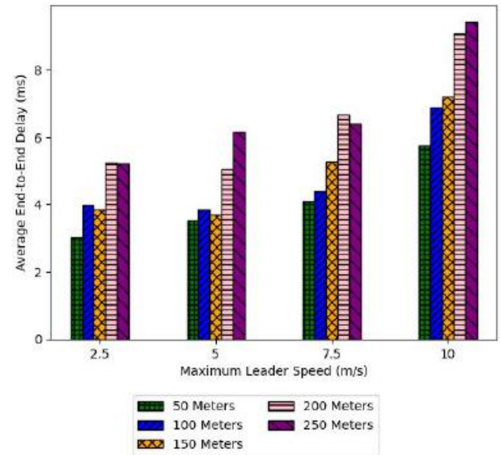

Fig. 7. Average End-to-End Delay for Varying Speed and Formation Sizes

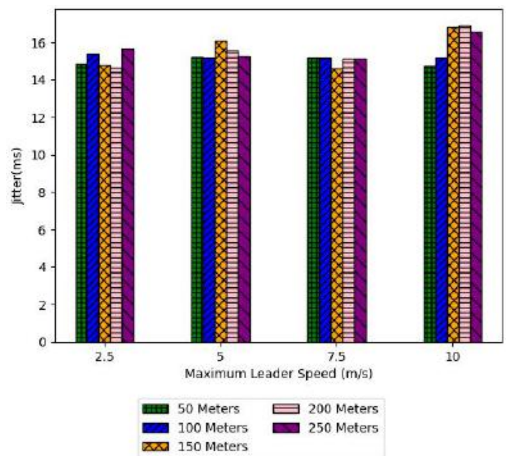

Fig. 8. Jitter incurred for Varying Leader Speed and Formation Sizes

end-to-end delay and jitter were incurred at a speed of 10m/s and formation size of 200 and 250 meters at about 9ms and 18ms, respectively. Figure 9 shows zero packet ratio for all speeds and formation sizes except for speed 10m/s where a slight increase was recorded. Therefore, our proposed protocol maintains a good *Quality of Service* (QoS) for varying leader speed and formation sizes. This performance is due to the leader's strategic position in the formation. Even at a formation distance of 250 Meters, the leader will still be within a good transmission range (∼145 Meters, use Eq. 6) with any of the followers.

$$L_{dist} = (follower_d * \sqrt{3})/3 \tag{6}$$

where $follower_d$ is the distance between the followers

VI. Conclusion

We proposed a learning-based formation control protocol for autonomous UAV-fleet. The protocol utilizes the Q-learning principle to control multiple UAVs deployed for SAR. The protocol leverages the RSSI values obtained by a UAV from its

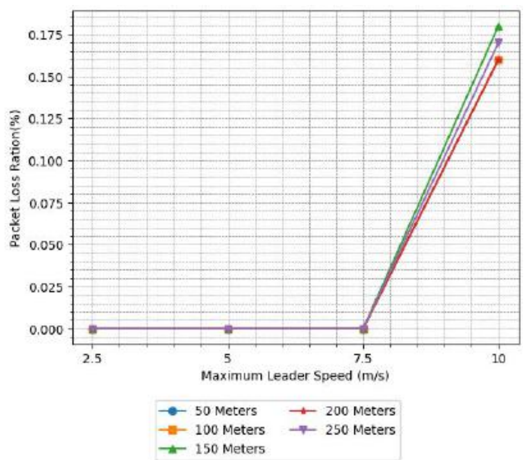

Fig. 9. Packet Loss Ratio of Video Packets for Varying Leader Speed and Formation Sizes

neighbors during wireless ad-hoc communication to maintain formation. Our protocol employs two Q-learning algorithms to control the followers autonomously: QSCA and DDFA. The QSCA is triggered to control the speed of the followers relative to that of the leader to ensure consistency of the topology; DDFA aligns the followers towards the direction of flight of the leader to ensure proper orientation of the formation. We conducted simulation experiments to evaluate the proposed protocol's performance to assess the fleet's formation stability and QoS of the video packets under varying leader speeds and formation sizes. Results reveal that our protocol is stable under a maximum leader speed of up to 5m/s and a formation size of 200 Meters. Also, the protocol incurred a maximum end-to-end delay of 9ms and jitter of 18ms, which are within the tolerated range.

In the future, we plan to extend this work to incorporate formation reconstruction in the event of the failure of any UAV during the mission and to conduct an extensive performance analysis of the protocol under a more complex search environment and simulation parameters.

References

[1] J. ZHANG and J. XING, "Cooperative task assignment of multi-uav system," *Chinese Journal of Aeronautics*, vol. 33, no. 11, p. 2825–2827, 2020.

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

[3] Y. Wu, J. Gou, X. Hu, and Y. Huang, "A new consensus theory-based method for formation control and obstacle avoidance of uavs," *Aerospace Science and Technology*, vol. 107, p. 106332, 2020.

[4] M. R. Brust, M. Zurad, L. Hentges, L. Gomes, G. Danoy, and P. Bouvry, "Target tracking optimization of uav swarms based on dual-pheromone clustering," *2017 3rd IEEE International Conference on Cybernetics (CYBCONF)*, 2017.

[5] R. Wise and R. Rysdyk, "Uav coordination for autonomous target tracking," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006.

[6] S. Drake, K. Brown, J. Fazackerley, and A. Finn, "Autonomous control of multiple UAVs for the passive location of radars," *2005 International Conference on Intelligent Sensors, Sensor Networks, and Information Processing*, 2005.

[7] J. Zhang, J. Yan, and P. Zhang, "Multi-uav formation control based on a novel back-stepping approach," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, p. 2437–2448, 2020.

[8] N. R. Zema, D. Quadri, S. Martin, and O. Shrit, "Formation control of a mono-operated uav fleet through ad-hoc communications: A q-learning approach," *2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2019.

[9] G. Djuknic and R. Richton, "Geolocation and assisted gps," *Computer*, vol. 34, no. 3, p. 123–125, 2001.

[10] K. Kaemarungsi and P. Krishnamurthy, "Modeling of indoor positioning systems based on location fingerprinting," *IEEE INFOCOM 2004*.

[11] P. Bahl and V. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*.

[12] G.-H. Kim, J.-C. Nam, I. Mahmud, and Y.-Z. Cho, "Multi-drone control and network self-recovery for flying ad hoc networks," *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*, 2016.

[13] N. Güzey, "Rf source localization using multiple uavs through a novel geometrical rssi approach," *Drones*, vol. 6, no. 12, p. 417, 2022.

[14] N. Guzey, H. M. Guzey, and A. Ronzhin, "Consensus-based localization by using array of antennas on a fixed-wing uav," *2019 27th Telecommunications Forum (TELFOR)*, 2019.

[15] L. Ruetten, P. A. Regis, D. Feil-Seifer, and S. Sengupta, "Area-optimized uav swarm network for search and rescue operations," *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, 2020.

[16] D. Ebrahimi, S. Sharafeddine, P.-H. Ho, and C. Assi, "Autonomous uav trajectory for localizing ground objects: A reinforcement learning approach," *IEEE Transactions on Mobile Computing*, vol. 20, no. 4, p. 1312–1324, 2021.

[17] K. Choutri, M. Lagha, and L. Dala, "A fully autonomous search and rescue system using quadrotor uav," *IJCDS Journal*, vol. 10, pp. 403–414, 04 2021.

[18] J. Klaue, B. Rathke, and A. Wolisz, "Evalvid – a framework for video transmission and quality evaluation," *Computer Performance Evaluation. Modelling Techniques and Tools*, p. 255–272, 2003.

[19] D. Saladino, A. Paganelli, and M. Casoni, "A tool for multimedia quality assessment in ns3: Qoe monitor," *Simulation Modelling Practice and Theory*, vol. 32, pp. 30–41, 2013.