Multi-Target Tracking Using a Swarm of UAVs by Q-learning Algorithm

Seyed Ahmad Soleymani[1], Shidrokh Goudarzi[2], Xingchi Liu[3],
Lyudmila Mihaylova[3], Wenwu Wang[1], and Pei Xiao[4]

[1]Centre for Vision Speech and Signal Processing (CVSSP), The University of Surrey, UK
[2]School of Computing and Engineering, University of West London, UK
[3]Department of Automatic Control and Systems Engineering, The University of Sheffield, UK
[4]Institute for Communication Systems (5GIC), The University of Surrey, UK

***Abstract*—This paper proposes a scheme for multiple unmanned aerial vehicles (UAVs) to track multiple targets in challenging 3-D environments while avoiding obstacle collisions. The scheme relies on Received-Signal-Strength-Indicator (RSSI) measurements to estimate and track target positions and uses a Q-Learning (QL) algorithm to enhance the intelligence of UAVs for autonomous navigation and obstacle avoidance. Considering the limitation of UAVs in their power and computing capacity, a global reward function is used to determine the optimal actions for the joint control of energy consumption, computation time, and tracking accuracy. Extensive simulations demonstrate the effectiveness of the proposed scheme, achieving accurate and efficient target tracking with low energy consumption.**

***Index Terms*—Multi-target tracking, UAV, Q-Learning, Edge Computing.**

I. Introduction

Unmanned Aerial Vehicles (UAVs) have emerged as a highly promising platform for target tracking systems, primarily owing to their exceptional mobility, adaptable deployment capabilities, and cost-effectiveness [1]. The versatility of UAVs lies in their ability to cover vast areas across different altitudes and locations, while also offering superior Line-of-Sight (LoS) links compared to ground Base Stations (BSs), courtesy of their elevated altitude. Consequently, UAVs stand as an ideal choice for target tracking applications. Especially in challenging scenarios where ground service agents are unavailable, UAVs play a pivotal role, diligently and precisely tracking targets [2].

However, limited communication range, battery capacity, and computing capacity are the main challenges of UAVs in a target tracking system. To deal with these challenges, a swarm of autonomous UAVs can be effective. A swarm of UAVs can be used to ensure effective communication coverage in the long term. The utilization of Edge Computing (EC) is also a promising solution to tackle the challenges faced by UAVs. For example, by leveraging the computational capacity of the edge, compute-intensive operations of UAVs can be offloaded to Edge Nodes (ENs) and as a result, enhance both computing quality and the lifetime of the UAVs network [3], [4]. As shown in [5], UAV-enabled EC has been conceptualized as a viable option to enhance the target tracking process.

In recent years, UAV-aided target detection and tracking has been studied. In [6], a Deep Q-Network (DQN) was constructed, with a finite action space, to deal with the limited field of view (FOV) of the camera equipped on the UAV, where a reward function was designed to take into account whether a target is within the FOV. In [7], authors introduced a motion planning algorithm based on the unscented Kalman filter (UKF) for UAVs to estimate the state of the target. The motion planner determines the UAV trajectory, which includes acceleration and turn rate. In [8], a reinforcement learning (RL) technique is used to train a swarm of UAVs to determine the optimal routes that maximize the probability of observing the targets. Existing works on target tracking employed different technologies and methods. However, it is still an open research problem. According to [9], mobile target tracking is a challenging problem due to the uncontrollable motion of the target, making the task even more complicated.

In this work, we focus on addressing the challenge of controlling multiple UAVs to track multiple targets, with the constraints of communication and computing resources of UAVs. To this end, we present a new approach where RSSI is used, due to its low cost and power consumption, hardware simplicity, and the ability to use simple receivers. More specifically, a Q-learning-based algorithm for UAV control action selection is proposed, along with a novel reward function that encourages UAVs to learn an optimal policy for improved tracking with maximum expected cumulative reward while considering accuracy, latency, and energy consumption. The key contributions of the paper are as follows:

1- We present a scheme using the QL algorithm that controls multiple UAVs in 3-D environments to achieve optimal tracking of multiple targets.

2- We develop an efficiency-maximizing reward function that accounts for joint optimization of accuracy, delay, energy consumption, and obstacle avoidance.

The paper is organized as follows: Section II presents the system model, Section III explains the proposed scheme, Section IV analyzes the scheme through simulations, and Section V provides concluding remarks.

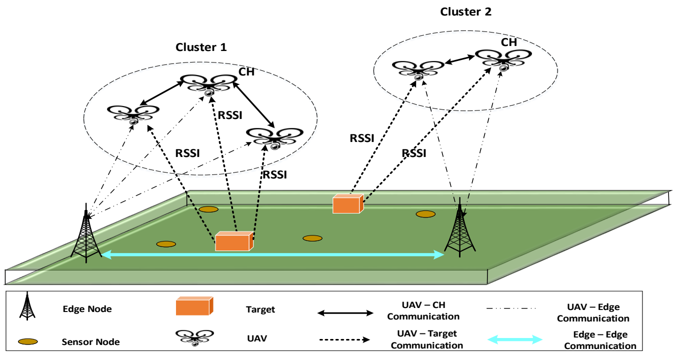


Fig. 1. Network Model.

II. System Model

In this section, we discuss the target and UAV trajectory models along with the channel model between UAVs and the target, for the scenario shown in Fig. 1, which includes the targets, ENs, and UAVs equipped with RSS sensors.

A. Target Trajectory Model

In the system, there are M targets that have mobility on the ground. Each target has a start point (x_m^s, y_m^s) and endpoint (x_m^e, y_m^e), where $m = 1, \cdots, M$. Each target chooses a path between these two points for its movement by considering obstacle avoidance. The initial location of m-th Radio Frequency (RF) target is fixed at $pos_m^{tar} = [x_m^{tar}(0) = x_m^s, y_m^{tar}(0) = y_m^s]$ and the time-varying location of target is denoted as $pos_m^{tar}(t) = [x_m^{tar}(t), y_m^{tar}(t)]$ at time t. Here, the target movement velocity is defined as $v_m^{tar}(t) = \left[v_{x,m}^{tar}(t), v_{y,m}^{tar}(t)\right]$.

B. UAV Trajectory Model

In this system, there exists N UAVs in which each UAV flies at different altitudes. We assume that the initial location of the UAV at time $t = 0$ is $pos_n^{uav}(0) = [x_n^{uav}(0), y_n^{uav}(0), z_n^{uav}(0)]$, where $n = 1, \cdots, N$. The time-varying location of the n-th UAV at time t is denoted as $pos_n^{uav}(t) = [x_n^{uav}(t), y_n^{uav}(t), z_n^{uav}(t)]$ and flight velocity of UAV is defined as $v_n^{uav}(t) = \left[v_{x,n}^{uav}(t), v_{y,n}^{uav}(t), v_{z,n}^{uav}(t)\right]$. Let $pos_n^{uav}(t)$ be the coordinate of the n-th UAV at time t. Hence, the sequence of points $L_n = \{pos_n^{uav}(0), \cdots, pos_n^{uav}(T_n)\}$ can be used to express the trajectory of the n-th UAV where T_n is the total time that n-th UAV flies during its trajectory, which depends on the trajectory length and velocity of the UAV, and can be obtained as follows [10]:

$$T_n = \sum_{t=0}^{T-1} \frac{\|pos_n^{uav}(t+1) - pos_n^{uav}(t)\|}{v_n^{uav}(t+1)} \tag{1}$$

C. Channel Model

The received power captured by the RSS sensor mounted on the n-th UAV at time t can be mathematically expressed as [11]:

$$rssi_n^{uav}(t) = P_{TX} - PL_n(t) - \rho_n, \tag{2}$$

here, P_{TX} represents the constant transmit power of the RF target, while $PL_n(t)$ denotes the path loss between the n-th UAV and the target at time t. ρ_n is an exponential random variable with a unit mean incorporating the effect of Rayleigh fading. The RSS measurements in each UAV can be denoted by $RSSI_n = [rssi_n^{uav}(0), \cdots, rssi_n^{uav}(T_n)]$.

III. Design of Multi-Target Tracking by Multi-UAV based on Q-Learning and Multilateration

In this section, we outline our scheme for the multi-target tracking problem. In this work, Q-learning, normalization, and multilateration form the core of our scheme.

A. Q-Learning

The Q-learning algorithm is a value-based Reinforcement Learning (RL) technique that is specifically designed for deterministic policies. In RL algorithms, the primary goal is to identify the optimal policy π^* that maximizes the cumulative reward over the long term. During each time slot, the QL algorithm determines an action to be performed by the UAV. Upon taking an action a, the UAV receives a reward $r(s,a)$ and transitions to a new state s'. Following each decision, the Q-value of the state-action pair is updated as:

$$Q(s,a) \leftarrow (1-\alpha) Q(s,a) + \alpha \left[r(s,a) + \gamma \max_{\mathrm{a'\in A}} Q(s',a') \right] \tag{3}$$

where $\gamma \in (0,1]$ is a discount factor that determines the importance of future rewards, and α is the learning rate that controls the extent to which new information overrides old information. The optimal policy can be learned through interactions with the environment and recording the corresponding experiences (s,a,r,s').

B. Multilateration

Multilateration is the process of determining the unknown position coordinates of a point of interest. In target tracking, using multilateration method for locating the m-th target with position pos_m^{tar}, the distance from $r_{m,n}$ to n-th UAV with position pos_n^{uav} is given as

$$r_{m,n} = \sqrt{\left(x_m^{tar} - x_n^{uav}\right)^2 + \left(y_m^{tar} - y_n^{uav}\right)^2} \tag{4}$$

C. Multi Target Tracking Using a Swarm of UAVs

In this work, a swarm of UAVs was considered to track each target. Once the position of the detected target is estimated, the edge node (EN) selects a swarm of nearby UAVs to track the target. These UAVs form a cluster consisting of a Cluster Head (CH) and other UAVs that are directly and wirelessly connected to the CH. Since each UAV is limited by its battery capacity, the EN selects a UAV with the highest battery capacity as the CH. It is worth noting that the number of UAVs in each cluster should be at least two.

Since the Q-learning algorithm utilized in UAVs is a state-action algorithm, we considered some allowable control actions for UAVs that can be taken at each state. In this work, the number of actions is equal to $\eta = 8$. These actions denote the flight direction along the x, y, and z-axis. UAVs determine the flight direction by choosing one action from discrete action space $AS = \{a_1, a_2, \cdots, a_\eta\}$. We assumed that UAVs have only horizontal movement, hence, the UAV dynamics are formulated as follows:

$$pos_n^{uav}(t) = pos_n^{uav}(t-1) + \begin{bmatrix} d * cos(\theta_i) \\ d * sin(\theta_i) \\ z_n^{uav}(t-1) \end{bmatrix} \quad (5)$$

where d is the velocity of the target at time t, $\theta_i = i * \frac{2\pi}{|AS|}$ for $i \in [1, \eta]$.

The Q-learning algorithm considers three parameters, namely accuracy, delay, and energy, to optimize the target tracking performance of UAVs. To account for these parameters, we designed a reward function that aims to minimize energy and delay while maximizing accuracy. Thus, the reward function can be expressed as follows:

$$reward = w_1 * (1 - E^*) + w_2 * (1 - D^*) + w_3 * A^* \quad (6)$$

The weights assigned to energy, delay, and accuracy are denoted by w_1, w_2, and w_3, respectively. The normalized values of consumed energy, delay, and accuracy, obtained through Min-Max normalization [12], are represented by E^*, D^*, and A^*, respectively.

In this work, the energy consumption of UAVs E is determined by the energy consumed during flight of UAV E_{flight} as follows [13]:

$$E = E_{flight} \quad (7)$$

where

$$E_{flight} = (W_{uav} \times g \times dist) + (F_p \times v_n^{uav} \times dist) \quad (8)$$

Here, E_{flight} is the total energy consumption during flight (in joules), W_{uav} is the weight of the UAV (in kilograms), $g = 9.81 m/s^2$ is the acceleration due to gravity, $dist$ is the total distance traveled during the flight (in meters), F_p is the average propulsion force required to maintain flight (in newtons), and v_n^{uav} is the average flight speed (in meters per second).

The delay between the target and UAV is directly proportional to the time taken for the signal to propagate between them. Hence, we can express the delay D as a function of propagation time D_{prop} as follows:

$$D = D_{prop} \quad (9)$$

where $D_{prop} = Distance/Speed$, $(Speed = 3 \times 10^8 m/s)$. As the distance between the target and UAV increases, the propagation time also increases, leading to an increase in the overall delay.

To compute accuracy A, the distance between UAV and the target is considered as follows:

$$A = dist_{mn} = \left\| pos_n^{uav} - pos_m^{tar} \right\| \quad (10)$$

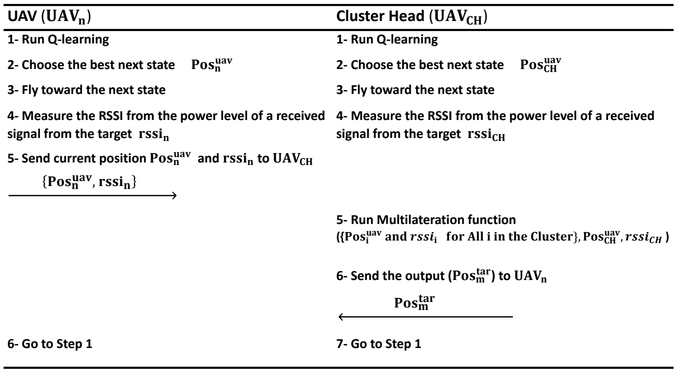
where $\|.\|$ is the Euclidean distance.

The parameters E, D, and A are measured and then subjected to Min-Max normalization to accommodate their different ranges of values and units. The normalized values are denoted as E^*, D^*, and A^* in the output. Additionally, since accuracy is considered more important than energy consumption and delay, we assigned it a higher weight. Specifically, we set $w_1 = \frac{1}{4}$, $w_2 = \frac{1}{4}$, and $w_3 = \frac{1}{2}$.

In each state, every UAV selects the optimal action from a set of η possible actions (i.e., flight directions) using the Q-learning algorithm and leveraging the reward function. To achieve the final objective, avoiding obstacle collisions, we assign a reward value of $reward = 0$ to each action where the probability of obstacle collision is high. Then, UAV measures the RSSI from the power level of a received signal of the target m. The RSSI $rssi_n^{uav}(t)$ measured by UAV n as well as current position $pos_n^{uav}(t)$ of UAV available in the cluster will be sent to the CH. Next, CH executes the Multilateration function and estimates the position of the target m. Finally, CH sends the estimated position to all UAVs in the cluster. Once UAVs receive the position of the target, UAVs run the Q-learning algorithm for selecting the next state. This process will be repeated until the target (e.g. m) reaches the endpoint (e.g. (x_m^e, y_m^e)). Fig. 2 represents the process of target position estimation as well as target tracking by our scheme. This figure also shows the process of data communication between a UAV and a CH.

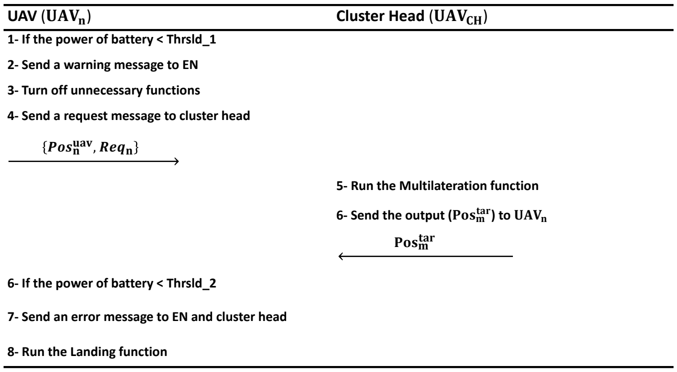
As mentioned above, each UAV is limited by its battery capacity. The energy consumption of each UAV is affected by several factors such as weight, aerodynamics, flight speed and altitude, and environmental conditions. Computation, communication, and task complexity also contribute to power consumption. The UAV's onboard computing system includes a processor, memory, and other components that consume energy. The processing load is primarily determined by the task complexity, dataset size, and algorithm used. Communication between UAVs and ground stations requires the use of communication systems, such as radios or transceivers, which also consume energy. The energy consumption of the communication system depends on the amount of data transmitted or received, the distance, and the quality of the communication link. Higher data rates or longer distances typically require higher transmit powers, leading to higher energy consumption.

To address this issue, we established two threshold values for the battery power of UAVs. These thresholds are utilized to monitor the state of the battery during the UAV's flight. By setting these threshold values, we can ensure that the UAV operates within a predetermined energy budget, which not only prolongs its flight time and range but also enhances its reliability and lowers the likelihood of battery depletion during a mission. Whenever the battery power of a UAV, such as UAV_n, falls below the first threshold value, it sends a warning message to the nearby EN to report its status. It also stops measuring RSSI and sends a request message to CH asking for the target's position until its battery power is sufficient for target tracking. Concurrently, the EN attempts to find a replacement UAV to swap with UAV_n. If UAV_n's battery power falls below the second threshold value, it sends



UAV (UAV_n)	Cluster Head (UAV_{CH})
1- Run Q-learning	1- Run Q-learning
2- Choose the best next state Pos_n^{uav}	2- Choose the best next state Pos_{CH}^{uav}
3- Fly toward the next state	3- Fly toward the next state
4- Measure the RSSI from the power level of a received signal from the target $rssi_n$	4- Measure the RSSI from the power level of a received signal from the target $rssi_{CH}$
5- Send current position Pos_n^{uav} and $rssi_n$ to UAV_{CH} $\xrightarrow{\{Pos_n^{uav}, rssi_n\}}$	
	5- Run Multilateration function $(\{Pos_i^{uav}$ and $rssi_i$ for All i in the Cluster$\}, Pos_{CH}^{uav}, rssi_{CH})$
	6- Send the output (Pos_m^{tar}) to UAV_n $\xleftarrow{Pos_m^{tar}}$
6- Go to Step 1	7- Go to Step 1

Fig. 2. The process of data communication between UAV and CH.



UAV (UAV_n)	Cluster Head (UAV_{CH})
1- If the power of battery < Thrsld_1	
2- Send a warning message to EN	
3- Turn off unnecessary functions	
4- Send a request message to cluster head $\xrightarrow{\{Pos_n^{uav}, Req_n\}}$	
	5- Run the Multilateration function
	6- Send the output (Pos_m^{tar}) to UAV_n $\xleftarrow{Pos_m^{tar}}$
6- If the power of battery < Thrsld_2	
7- Send an error message to EN and cluster head	
8- Run the Landing function	

Fig. 3. The process of UAV battery power monitoring.

an error message to the nearby EN and CH, and then initiates the landing function. This process is illustrated in Fig. 3.

IV. Numerical Results

This section presents the numerical results of our scheme, specifically tracking accuracy and energy consumption. MATLAB was used as the simulation platform, and an obstacle-filled environment was created using a matrix with cylinders and cones representing the obstacles. The simulation involved five UAVs tracking two targets in this environment. Here, the tracking of target 1 is performed by three UAVs, namely $\{UAV_1, UAV_2, UAV_3\}$, while target 2 is tracked by two UAVs, namely $\{UAV_4, UAV_5\}$. We also included three edge nodes (ENs) in the simulation. Each UAV is capable of communicating with an EN that is within its communication range. In order to assess the effectiveness of our scheme, we established three separate scenarios, outlined as follows:

- **Cluster 1**: In this scenario, three UAVs are organized into a cluster, and a single UAV is designated as the cluster head (CH). The two remaining UAVs communicate with the CH and nearby EN and do not directly communicate with each other.
- **Cluster 2**: In this scenario, two UAVs are grouped into a cluster, and one UAV is elected as a cluster head (CH). Another UAV is able to communicate with CH and nearby ENs.
- **Non-Clustered**: In this scenario, there is no clustering of UAVs. Instead, three individual UAVs are assigned to track a target and are able to communicate with each other as well as nearby ENs.

The initial positions of each UAV and target were defined as previously explained. Target 1 has a starting position of $pos_1^{tar}(0) = [2, 1]$ meters, while target 2 has a starting position of $pos_2^{tar}(0) = [1, 6]$ meters. Both targets have a designated endpoint of $[30, 15]$ meters. To move toward the endpoint while avoiding obstacles, each target randomly selects a path between its start point and the endpoint. Additionally, we have defined the initial positions of five UAVs as $pos_1^{uav}(0) = [1, 2, 2.5]$, $pos_2^{uav}(0) = [3, 4.5, 3]$, $pos_3^{uav}(0) = [6, 1, 2]$, $pos_4^{uav}(0) = [4.5, 6, 3]$, and $pos_5^{uav}(0) = [2, 10, 4]$ meters. Each target is initially assigned a velocity, and their velocities can vary from $1m/s$ to $5m/s$ during their movement along the trajectory. The UAVs adjust their velocity during target tracking based on the velocity of the target. Here, we assume that each UAV will receive information about obstacles from nearby ENs to avoid the collision. The information includes the dimensions of the obstacles such as length, width, height, diameter, and other relevant details.

The root means square error (RMSE) can be an effective metric for assessing the accuracy of the scheme's performance [14]. To this end, we consider the actual position of the target and the estimated position of the target by the UAVs. For all positions that the target passed during its trajectory, we measured the RMSE. We carried out this procedure for each of the aforementioned scenarios individually. The RMSE was computed using the following equation:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{K}(x_i^{tar} - \hat{x}_i^{tar})^2 + (y_i^{tar} - \hat{y}_i^{tar})^2}{K}} \quad (11)$$

where K is the number of positions that the target passed during its trajectory, and $[x_i^{tar}, y_i^{tar}]$ and $[\hat{x}_i^{tar}, \hat{y}_i^{tar}]$ represent the actual and estimated positions of the target at the i-th position, respectively. Fig. 4 presents a comparison of the RMSE for each scenario. It is observed that the accuracy of the proposed scheme in scenario 1 is superior to those in other scenarios. This can be attributed to the higher number of UAVs present in scenario 1, as compared to scenario 2 in cluster-based scenarios. In real-time applications like target-tracking, both the computation and communication delay have a significant impact on application performance accuracy. The reduced number of connections and communications for sending/receiving information between UAVs in scenario 1 compared to scenario 3 leads to higher accuracy in the former.

We conducted experiments to measure the total energy consumption by each UAV during the target tracking process. The results, as shown in Fig. 5, indicate that the energy consumed by UAVs in the cluster-based scenario is less than that in the non-clustered scenario. This is due to the reduced communication and computation requirements in the cluster-based scenario.

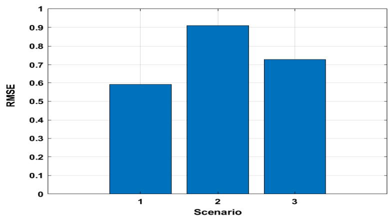


Fig. 4. Comparison of measured RMSE in each scenario

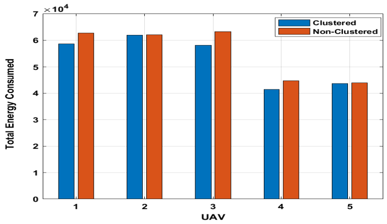


Fig. 5. Comparison of total energy consumption by each UAV in clustered and non-clustered scenarios

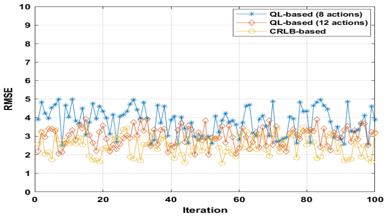


Fig. 6. Comparison of our scheme with CLRB-based scheme over 100 Monte Carlo experiments.

In addition, we conduct a comparative analysis of our scheme with a Cramér–Rao Lower Bound (CRLB) based scheme proposed in [15] over 100 Monte Carlo experiments. The CRLB is a fundamental concept used in target tracking to estimate the accuracy of any unbiased estimator. It serves as a benchmark for assessing the quality of target tracking algorithms. The comparison focuses on assessing the performance in terms of Root Mean Squared Error (RMSE). For our scheme, we evaluate its performance under the first scenario with different numbers of allowable control actions ($\eta = 8$ and 12). As depicted in Fig. 6, the QL-based control demonstrates tracking performance comparable to the CRLB-based control, which is considered the optimal control scheme.

V. Conclusion

In this study, a scheme based on RSSI has been proposed for tracking multiple targets using multiple UAVs. The QL algorithm and Multilateration are the core of the proposed scheme. Due to the limitation of power capacity and the computing capacity of UAVs and in addition, the importance of delay in the target tracking, energy consumption, delay, and accuracy have been considered as three main parameters in the reward function of the QL algorithm. We have analyzed our scheme in cluster-based and non-cluster-based scenarios. The obtained results showed that our scheme based on clustering has provided a more accurate and efficient target-tracking solution with lower energy.

Acknowledgment

This research was sponsored by the US Army Research Laboratory and the UK MOD University Defence Research Collaboration (UDRC) in Signal Processing and was accomplished under Cooperative Agreement Number W911NF-20-2-0225. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory, the MOD, the U.S. Government, or the U.K. Government. The U.S. Government and U.K. Government are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] J. Wang, C. Jiang, Z. Han, Y. Ren, R. G. Maunder, and L. Hanzo, "Taking drones to the next level: Cooperative distributed unmanned-aerial-vehicular networks for small and mini drones," *IEEE VehIcular Technology Magazine*, vol. 12, no. 3, pp. 73–82, 2017.

[2] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs," *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 3949–3963, 2016.

[3] S. Goudarzi, M. H. Anisi, H. Ahmadi, and L. Musavian, "Dynamic resource allocation model for distribution operations using sdn," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 976–988, 2020.

[4] S. Goudarzi, S. A. Soleymani, W. Wang, and P. Xiao, "Uav-enabled mobile edge computing for resource allocation using cooperative evolutionary computation," *IEEE Transactions on Aerospace and Electronic Systems*, 2023.

[5] J. Wang, K. Liu, and J. Pan, "Online uav-mounted edge server dispatching for mobile-to-mobile edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1375–1386, 2019.

[6] S. Bhagat and P. Sujit, "Uav target tracking in urban environments using deep reinforcement learning," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2020, pp. 694–701.

[7] L. Wang, Y. Li, H. Zhu, and L. Shen, "Target state estimation and prediction based standoff tracking of ground moving target using a fixed-wing uav," in *IEEE ICCA 2010*. IEEE, 2010, pp. 273–278.

[8] T. Wang, R. Qin, Y. Chen, H. Snoussi, and C. Choi, "A reinforcement learning approach for uav target searching and tracking," *Multimedia Tools and Applications*, vol. 78, pp. 4347–4364, 2019.

[9] X. Deng, J. Li, P. Guan, and L. Zhang, "Energy-efficient uav-aided target tracking systems based on edge computing," *IEEE Internet of Things Journal*, vol. 9, no. 3, pp. 2207–2214, 2021.

[10] Y.-J. Chen, D.-K. Chang, and C. Zhang, "Autonomous tracking using a swarm of uavs: A constrained multi-agent reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13 702–13 717, 2020.

[11] F. Shang, W. Su, Q. Wang, H. Gao, and Q. Fu, "A location estimation algorithm based on rssi vector similarity degree," *International Journal of Distributed Sensor Networks*, vol. 10, no. 8, p. 371350, 2014.

[12] M. F. Aslan, A. Durdu, and K. Sabanci, "Visual-inertial image-odometry network (viionet): A gaussian process regression-based deep architecture proposal for uav pose estimation," *Measurement*, vol. 194, p. 111030, 2022.

[13] T. Zhang, Y. Xu, J. Loo, D. Yang, and L. Xiao, "Joint computation and communication design for uav-assisted mobile edge computing in iot," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5505–5516, 2019.

[14] S. Goudarzi, S. Ahmad Soleymani, M. H. Anisi, D. Ciuonzo, N. Kama, S. Abdullah, M. Abdollahi Azgomi, Z. Chaczko, and A. Azmi, "Real-time and intelligent flood forecasting using uav-assisted wireless sensor network," *Computers, Materials and Continua*, vol. 70, no. 1, pp. 715–738, 2021.

[15] S. Papaioannou, S. Kim, C. Laoudias, P. Kolios, S. Kim, T. Theocharides, C. Panayiotou, and M. Polycarpou, "Coordinated crlb-based control for tracking multiple first responders in 3d environments," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2020, pp. 1475–1484.