Deep Reinforcement Learning Multi-UAV Trajectory Control for Target Tracking

Jiseon Moon, Savvas Papaioannou, *Member, IEEE*, Christos Laoudias, Panayiotis Kolios and Sunwoo Kim, *Senior Member, IEEE*

***Abstract*—In this paper we propose a novel Deep Reinforcement Learning (DRL) approach for controlling multiple Unmanned Aerial Vehicles (UAVs) with the ultimate purpose of tracking multiple First Responders (FRs) in challenging 3D environments in the presence of obstacles and occlusions. We assume that the UAVs receive noisy distance measurements from the FRs which are of two types i.e., Line of Sight (LoS) and Non-LoS (NLoS) measurements and which are used by the UAV agents in order to estimate the state (i.e., position) of the FRs. Subsequently, the proposed DRL-based controller selects the optimal joint control actions according to the Cramér-Rao Lower Bound (CRLB) of the joint measurement likelihood function to achieve high tracking performance. Specifically, the optimal UAV control actions are quantified by the proposed reward function which considers both the CRLB of the entire system and each UAV's individual contribution to the system, called global reward and difference reward, respectively. Since the UAVs take actions that reduce the CRLB of the entire system, tracking accuracy is improved by ensuring the reception of high quality LoS measurements with high probability. Our simulation results show that the proposed DRL-based UAV controller provides a highly accurate target tracking solution with a very low run-time cost.**

***Index Terms*—Multi-agent deep reinforcement learning, multi-target tracking, unmanned aerial vehicle**

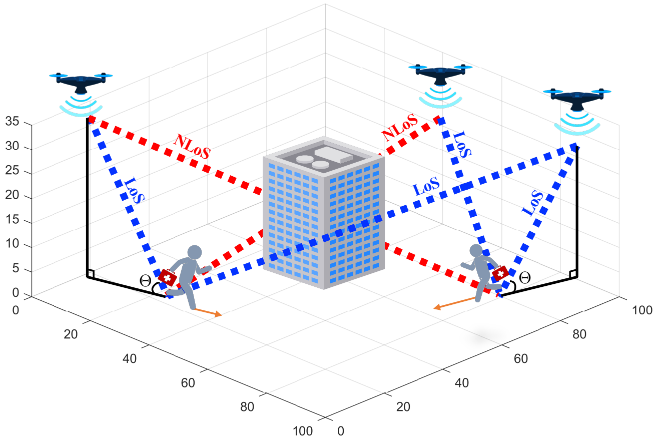


Fig. 1. An illustration of multiple UAVs system for first responders tracking. During the SAR mission, multiple UAVs track the first responders in a three-dimensional environment and receive noisy distance measurements from the first responders. UAVs adjust their position in order to estimate the state of first responders accurately.

I. Introduction

Nowadays, Unmanned Aerial Vehicles (UAVs) have become a promising technological platform offering high mobility, flexible deployment, and low cost [1, 2]. Thanks to the aforementioned advantages, UAVs are widely operated in various application scenarios such as wireless communication support [3–5], surveillance [6], delivery [7, 8] and Search And Rescue (SAR) [9–11]. That said, SAR missions could be extremely challenging and dangerous. Nowadays, SAR missions respond to devastations caused by floods, storms, maritime accidents, earthquakes, hazardous materials releases, etc. The first responders (FRs) often face various risky and dangerous situations, and they are required to work in areas where public services are unavailable and the infrastructure is destroyed and disrupted (e.g., during floods with downed power lines and gas leaks). Motivated by this, we believe that a team of autonomous mobile agents (e.g., UAVs) could become an important aid in many SAR missions by accurately tracking the FRs in the aforementioned challenging conditions. A robust and accurate multi-UAV tracking system for SAR missions not only can provide the required level of safety to the FRs but also allows for better organization and coordination of the rescue team, thus minimizing the need to place the rescuers in danger situations.

Various sensors, such as lidars and cameras, are nowadays mounted on UAVs to enable FRs with comprehensive environmental perception and help them successfully complete their missions. Cameras mounted on UAVs are mainly used to detect and track people, objects or natural disasters (e.g., wildfire, flood, earthquake). Frequently, the target position is detected and estimated by image processing and deep learning [12, 13] techniques. UAVs can also be equipped with RF sensors to provide, Received Signal Strength (RSS), Round-Trip Time (RTT), and Time of Arrival (TOA) type of measurements which are combined with filtering techniques to enable advanced navigation and target localization capabilities [14–17].

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2020-2017-0-01637) supervised by the IITP (Institute for Information & communications Technology Promotion) *(corresponding author : Sunwoo Kim)*

The work of Savvas Papaioannou, Christos Laoudias and Panayiotis Kolios is supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 739551 (KIOS CoE) and from the Republic of Cyprus through the Directorate General for European Programmes, Coordination and Development.

Jiseon Moon and Sunwoo Kim are with the Department of Electronic Engineering, Hanyang University, Seoul 04763, South Korea (e-mail: jiseonmoon@hanyang.ac.kr; remero@hanyang.ac.kr).

Savvas Papaioannou, Christos Laoudias and Panayiotis Kolios are with the KIOS Research and Innovation Center of Excellence, University of Cyprus (e-mail: papaioannou.savvas@ucy.ac.cy; laoudias@ucy.ac.cy; pkolios@ucy.ac.cy).

In this work, we are interested in the control of multiple UAVs for accurately tracking multiple FRs in the disaster environment. As illustrated in Fig. 1, we consider a scenario where a group of UAVs is used to track multiple FRs in the ground in order to assist the SAR mission. We assume that the FRs carry id-linked radio transmitters such as Bluetooth or Ultra-Wideband (UWB) tags. The UAVs receive the FRs radio transmissions and use this information to accurately localize them. We assume that the environment comprises of obstacles, occlusions and large structures, thus the measurements received by the UAVs at any time can be of two types namely Line of Sight (LoS) and Non-LoS (NLoS) measurements. The objective of this work is to control the UAV team operating in the aforementioned challenging conditions in order to provide optimized tracking of the FRs (i.e., the targets).

The target tracking performance can be quantified by the Cramér-Rao Lower Bound (CRLB), which is used in estimation theory to derive a lower bound of the variance of an unbiased estimator. In the localization system analysis, the CRLB implies that the localization error at a given position is greater than or equal to X meters given the conditions in the region of interest, including the number of the signal sources, the geometry of the RF receiver and the sources, and the statistical characteristics of the measurements [18]. In other words, the CRLB becomes larger (i.e., higher localization error should be expected) when fewer measurements are available to estimate the FRs position or when the geometry of the UAVs is not suitable. Hereafter, we intend to combine CRLB with Deep Reinforcement Learning (DRL) to decide the control actions of multiple UAVs in order to achieve accurate multi-target tracking.

Reinforcement Learning (RL) is a type of machine learning algorithm that considers how agents take decisions in an environment. By introducing the neural network as a function approximator in the training stage, DRL overcomes traditional RL shortcomings with a finite number of states and actions [19]. Interestingly, there have been several efforts regarding the application of DRL to multiple UAV systems in recent years, including network security [20], communication optimization [21–23], and target tracking and navigation [24–27]. Motivated by the recent advances in DRL techniques and applications, in this work we propose a novel control framework which combines the theory of DRL with the theory of state estimation through the utilization of CRLB. Specifically, we design a novel reward function for our DRL-based framework which in each time-step quantifies the achievable CRLB according to the applied joint UAV control actions. That said, through the agent-environment interaction, multiple UAVs learn an optimal policy that enhances the tracking performance and maximizes expected cumulative reward. Consequently, the UAVs adjust their trajectories to optimize the target state estimation by selecting the joint control actions, which achieve the lower CRLB.

The main contributions of this paper are the following:

- We propose the first DRL-CRLB based framework to control multiple UAVs in such a way so that multiple FRs are being optimally tracked in challenging 3D environments in the presence of both LoS and NLoS conditions.
- We design a novel reward function using the CRLB of the target state estimator. The total reward is composed of sub-rewards, which account for the CRLB of the entire system and the individual contribution of each UAV, thus enabling the proposed DRL framework to learn the optimal policy.
- We verify the proposed approach through extensive simulation experiments and compare it with existing solutions.

The rest of this paper is organized as follows. Section II reviews the related work on UAV control and target tracking. Section III introduces the background of the Markov Decision Process (MDP), DRL, and dueling network. Section IV describes the system model. The CRLB for FRs state estimator is presented in Section V. Section VI introduces our DRL-based multiple UAV control system for target tracking. Section VII presents the simulation results in the performance evaluation. Finally, Section VIII provides concluding remarks.

II. Related Work

In this section we summarize the most relevant works to the problem tackled in this paper. Specifically, we discuss the most recent UAV trajectory optimization and DRL-based techniques for UAV control, and we briefly summarize the main target tracking techniques, which have been used in related problems.

In recent years, some works on the trajectory optimization for the UAV-aided networks have been studied. The authors of [28, 29] propose a multiple rechargeable UAVs control technique in order to provide seamless and long-term services to the ground nodes. In [28], multiple UAVs adjust their trajectories, transmit power, and the node assignment by solving the UAVs configuration optimization problem, which is represented by the nonconvex problem. In [29], multiple UAVs determine their deployment and charging strategy using Discrete Particle Swarm Optimization (DPSO) algorithm. The authors of [30] propose the time-efficient UAV trajectory optimization techniques to collect traffic data from the roadside unis. To solve the problem, they introduce meta-heuristic methods Genetic Algorithm (GA) and harmonic search, and compare the performance of two methods.

A variety of approaches have been proposed for DRL-based UAV trajectory control to fit various applications. For instance, in order to secure the UAV transmitter against being wiretapped, UAV jammers send jamming signals to eavesdroppers by adjusting flying direction, transmit power level, and jamming power level [20]. By exploiting the dueling Deep Q-Network (DQN), multiple UAVs adjust their movement to maximize downlink capacity, covering ground terminals [21]. In [22, 23], an energy-efficient UAV control method is proposed. Each of the UAVs selects its flying direction and distance, considering the communications coverage, fairness, and connectivity. For the problem of UAVs control for target tracking and navigation, in [24], authors consider that persistent target tracking is a challenging task in an urban environment since a UAV equipped with a camera has a limited Field of View (FOV). They construct a DQN, called target following DQN, with finite action space and design a reward function that considers whether a target is within the FOV or not. In [25], a UAV path planning scheme is proposed for target tracking and obstacle avoidance. Deep Deterministic Policy Gradient (DDPG) allows a UAV to be operated in continuous action space by combining DQN with an actor-critic algorithm, which utilizes two networks (actor-network and critic-network) to determine the best action and evaluate the selected action, respectively. The reward function is designed considering the angle between the UAV and target and how smooth the UAV trajectory is. In [26, 27], DRL enables a single UAV to navigate from origin to destination by continuously controlling the UAVs' flight distance and direction. They consider the agent's state as sensory measurements, including angle and distance between the UAV's position and destination. Whenever a UAV executes the selected action, it receives a non-sparse reward, which considers the distance between the UAV and destination/obstacle [26]. On the other hand, a UAV only receives a sparse reward when it reaches its destination [27].

Regarding the single target tracking problem, authors in [15] propose a UAV motion planning algorithm for target's state estimation. The Unscented Kalman Filter (UKF) is used to estimate the target state, while UAV trajectory, including acceleration and turn rate, is determined by the motion planner. In [14], an Extended Kalman Filter (EKF)-based target tracking technique is proposed. A single UAV estimates a moving target state and then predicts the optimal trajectory from the estimated target's state. In the presence of multiple targets, recursive Bayesian filtering is used to formulate the multiple target searching and tracking problem [17]. Multiple UAVs manage their trajectories for searching and tracking, depending on whether the target is detected. In [16], the authors exploit the Gaussian Mixture Probability Hypothesis Density filter to estimate the number of targets and track target trajectories. This solution deals with complex environments, where the number of targets is unknown and varying.

In this work, we address the multiple UAVs control problem for tracking multiple targets. The proposed DRL-based controller constructs a deep dueling Q-network with continuous state space and discrete action space and applies a particle filter to estimate the multiple target states.

III. Background

Here, we introduce the background of MDP, DRL, and dueling network.

A. MDP Formulation

We model the proposed UAV control problem as MDP. An MDP is defined as a tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma\}$ which consists of five elements, i.e., states, actions, transition probabilities, rewards, and discount factor. The state in a state space $\mathcal{S}$ should be observable from the environment. The action in an action space $\mathcal{A}$ is determined by the agent's movement. The state and action space of MDP can be either continuous or discrete. In this paper, we consider the state space is continuous, and the action space is discrete. A set of state transition probability $\mathcal{P} = \{p(\mathbf{s}_{k+1} \mid \mathbf{s}_k, a_k) \mid \mathbf{s}_k,\ \mathbf{s}_{k+1} \in \mathcal{S},\ a_k \in \mathcal{A}\}$ is made up of the transition probability $p(\mathbf{s}_{k+1} \mid \mathbf{s}_k, a_k)$, which is defined by the distribution of the next state $\mathbf{s}_{k+1}$ given the current state $\mathbf{s}_k$ and taken action a_k. When the agent takes action a_k at the state $\mathbf{s}_k$, the agent receives a reward $r(\mathbf{s}_k, a_k)$ from a set of rewards for all possible state-action pairs $\mathcal{R} = \{r(\mathbf{s}_k, a_k) \mid \mathbf{s}_k \in \mathcal{S},\ a_k \in \mathcal{A}\}$. The last element $\gamma \in [0, 1]$ is the discount factor, which indicates the current value for the reward obtained in the future. A policy $\pi = p(a_k|\mathbf{s}_k)$ is a mapping from the agent's state to action and gives the probability of selecting a candidate action at the current state $\mathbf{s}_k$ [31].

The agent observes its state $\mathbf{s}_k \in \mathcal{S}$ from the environment and takes action $a_k \in \mathcal{A}$ according to policy. The interaction between agent and environment can be represented by trajectory $(\mathbf{s}_0,\ a_0,\ r_0,\ \mathbf{s}_1,\ a_1,\ r_1,\ \ldots)$. The cumulative discounted reward, called return, is given by:

$$G_k = R_k + \gamma R_{k+1} + \gamma^2 R_{k+2} + \cdots = \sum_{\tau=0}^{\infty} \gamma^\tau R_{k+\tau}. \tag{1}$$

The value function (state-value function) $V_\pi(\mathbf{s}, a)$ at state $\mathbf{s}$ is the expected return when the state is $\mathbf{s}$ under policy π, and it is given by:

$$V_\pi(\mathbf{s}) = \mathbb{E}_\pi[G_k \mid \mathbf{s}_k = \mathbf{s}] = \mathbb{E}_\pi\left[\sum_{\tau=0}^{\infty} \gamma^\tau R_{k+\tau} \mid \mathbf{s}_k\right]. \tag{2}$$

The Q-value function (action-value function) under policy π is defined as the expected return for taking action a in state $\mathbf{s}$, and it is represented as follows:

$$\begin{aligned} Q_\pi(\mathbf{s}, a) &= \mathbb{E}_\pi[G_k \mid \mathbf{s}_k = \mathbf{s}, a_k = a] & (3)\\ &= \mathbb{E}_\pi\left[\sum_{\tau=0}^{\infty} \gamma^\tau R_{k+\tau} \mid \mathbf{s}_k, a_k\right]. & (4) \end{aligned}$$

The value function and Q-value function have a relationship of $V_\pi(\mathbf{s}) = \mathbb{E}_{a\sim\pi(\mathbf{s})}[Q_\pi(\mathbf{s}, a)]$. The advantage function represents the importance of each action, defined by the value function and Q-value function. The advantage function subtracts the value function from the Q-value function $A_\pi(\mathbf{s}, a) = Q_\pi(\mathbf{s}, a) - V_\pi(\mathbf{s})$, and have a relationship of $\mathbb{E}_{a\sim\pi(\mathbf{s})}[A_\pi(\mathbf{s}, a)] = 0$.

B. Deep Reinforcement Learning

The objective of RL is to find an optimal policy π^* to maximize the Q-value function. The optimal Q-value function Q^* is the maximum expected return achievable from a given state-action pair, obtained using Bellman's optimality equation [32]:

$$\begin{aligned} Q^*(\mathbf{s}_k, a_k) &= \arg\max_\pi \mathbb{E}\left[\sum_{\tau\geq 0} \gamma^\tau R_{k+\tau} \mid \mathbf{s}_k, a_k, \pi\right] \\ &= \mathbb{E}\left[R + \gamma \max_{a_{k+1}} Q^*(\mathbf{s}_{k+1}, a_{k+1}) \mid \mathbf{s}_k, a_k, \pi\right]. \end{aligned} \tag{5}$$

For MDP with a finite number of states and actions, the optimal Q-value function can be approximated by updating Q-table iteratively, where rows represent the potential states, and columns represent actions.

However, the table-based Q learning is difficult to apply to large-scale problems with continuous state or action because of the memory capacity caused by a lot of states and actions. Likewise, our multiple UAVs control problem cannot be represented as table-based Q learning because the state space of multiple UAVs is infinite.

To solve this problem, DRL introduces a deep neural network $Q(\mathbf{s}_k, a_k; \theta)$ to approximate optimal Q-value function, where the parameter θ is the weights of a neural network, named Q-network. During DRL training stage, the agent's transition $(\mathbf{s}_k, a_k, r_k, \mathbf{s}_{k+1})$ is stored into a replay memory $\mathcal{D}$. To achieve sufficient learning, minibatches are randomly drawn from the replay memory to adjust Q-network weight rather than using the batches of consecutive samples. The Q-network is updated by minimizing the loss function, which is given by:

$$\mathrm{Loss}(\theta) = \mathbb{E}_{\mathbf{s}_k, a_k, r_k, \mathbf{s}_{k+1}} \left[(y_k - Q(\mathbf{s}_k, a_k; \theta))^2 \right], \tag{6}$$

with

$$y_k = r(\mathbf{s}_k, a_k) + \gamma \max Q^{-}(\mathbf{s}_{k+1}, a_{k+1}; \theta^{-}), \tag{7}$$

where the target value y_k is a summation of reward of state-action pair $r(\mathbf{s}_k, a_k)$ and the maximum discounted Q-value of the target network Q^{-}, which is parameterized by the weights of the target network θ^{-}. The weight of the target network θ^{-} is updated by Q-network θ every $\mathcal{N}$ timesteps.

C. Dueling Deep Q-network

In this paper, we utilize a dueling architecture to achieve robust estimates of Q-value function. The dueling architecture decouples Fully Connected (FC) layers into two streams rather than using a single stream of FC layers, i.e., original deep Q network. In dueling architecture, one stream of FC layers is a value function estimator $V(\mathbf{s}_k; \theta, \theta_\beta)$ that outputs a scalar, and the other stream is an advantage function estimator $A(\mathbf{s}_k, a_k; \theta, \theta_\alpha)$ that outputs a $|\mathcal{A}|$-dimensional vector. Here, θ_α and θ_β denote weights of the advantage function estimator and the value function estimator, respectively. According to the advantage function definition, these two streams are combined to calculate the Q-value function, as follows:

$$Q_\pi(\mathbf{s}, a) = V_\pi(\mathbf{s}) + A_\pi(\mathbf{s}, a). \tag{8}$$

However, $Q(\mathbf{s}_k, a_k; \theta, \theta_\alpha, \theta_\beta)$ is the only parameterized estimate of the Q-value function. Moreover, it is impossible to obtain $V_\pi(\mathbf{s})$ and $A_\pi(\mathbf{s}, a)$ uniquely for a given $Q_\pi(\mathbf{s}, a)$. To solve this issue, we modify the combination of the two streams to obtain the Q function as follows:

$$\begin{aligned} &Q(\mathbf{s}_k, a_k; \theta, \theta_\alpha, \theta_\beta) = \\ &V(\mathbf{s}_k; \theta, \theta_\beta) + \left(A(\mathbf{s}_k, a_k; \theta, \theta_\alpha) - \max_{a_k^{'} \in \mathcal{A}} A(\mathbf{s}_k, a_k^{'}; \theta, \theta_\alpha) \right). \end{aligned} \tag{9}$$

Due to the above modification, the advantage function estimator $A(\mathbf{s}_k, a_k; \theta, \theta_\alpha)$ has zero advantage for the selected action. Besides, for $a^* = \arg\max_{a' \in \mathcal{A}} Q(\mathbf{s}_k, a_k; \theta, \theta_\alpha, \theta_\beta) = \arg\max_{a' \in \mathcal{A}} A(\mathbf{s}_k, a_k; \theta, \theta_\alpha)$, we get $Q(\mathbf{s}_k, a_k^*; \theta, \theta_\alpha, \theta_\beta) = V(\mathbf{s}_k; \theta, \theta_\beta)$.

Alternatively, the Q-value function is obtained by replacing the max operator with an average as follows [33]:

$$\begin{aligned} &Q(\mathbf{s}_k, a_k; \theta, \theta_\alpha, \theta_\beta) = \\ &V(\mathbf{s}_k; \theta, \theta_\beta) + \left(A(\mathbf{s}_k, a_k; \theta, \theta_\alpha) - \frac{1}{|\mathcal{A}|} \sum_{a_k^{'} \in \mathcal{A}} A(\mathbf{s}_k, a_k^{'}; \theta, \theta_\alpha) \right). \end{aligned} \tag{10}$$

Hence, the stream $V(\mathbf{s}_k; \theta, \theta_\beta)$ of FC layers estimates the value function, and the other stream $A(\mathbf{s}_k, a_k; \theta, \theta_\alpha)$ provides the estimate of the advantage function.

IV. System model

In this section we outline the modeling assumptions used in the proposed framework. In particular we describe the first responder dynamic model, UAV dynamic model, and the UAV sensing model.

A. First Responder Dynamics

We assume that during a SAR mission there are N (where N is known and fixed) first responders (i.e., targets) on the ground that need to be tracked. At timestep k, the state vector of the j-th target is represented by:

$$\mathbf{x}_k^j = [x^j, \dot{x}^j, \ddot{x}^j, y^j, \dot{y}^j, \ddot{y}^j, z^j, \dot{z}^j, \ddot{z}^j]_k^\intercal, \tag{11}$$

where x^j, y^j, z^j are Cartesian coordinates of the j-th target position, $\dot{x}^j, \dot{y}^j, \dot{z}^j$ denotes the speed of the j-th target along the x, y, and z direction, and finally $\ddot{x}^j, \ddot{y}^j, \ddot{z}^j$ is the acceleration of the j-th target along the x, y, and z direction in three-dimensional space.

During their operations the FRs encounter sudden and unexpected changes in their motion patterns. In order to account for this uncertainty, the dynamic model of the FRs is composed of a command process vector $\boldsymbol{\nu}_k = [\nu_x, \nu_y, \nu_z]_k^\intercal$ and a random acceleration vector $\mathbf{b}_k = [\ddot{x}, \ddot{y}, \ddot{z}]_k^\intercal$, where the total acceleration is $\mathbf{a}_k = \boldsymbol{\nu}_k + \mathbf{b}_k$. The command processes $\nu_{x,k}$, $\nu_{y,k}$ and $\nu_{z,k}$ take values from each set of the discrete acceleration level $\mathcal{L}_x$, $\mathcal{L}_y$ and $\mathcal{L}_z$. The command process vector $\boldsymbol{\nu}_k$ is formulated as a Markov chain with a set of finite states $\mathbb{L} = \mathcal{L}_x \times \mathcal{L}_y \times \mathcal{L}_z = \{v_1, \ldots, v_L\}$ and transition probability $L_{l\bar{l}} = p(\boldsymbol{\nu}_k = v_{\bar{l}} \mid \boldsymbol{\nu}_{k-1} = v_l)$, $l, \bar{l} = \{1, \ldots, L\}$. The transition probability is given by:

$$L_{l\bar{l}} = \begin{cases} p_l, & \text{if } l = \bar{l} \\ (1 - p_l)/(L-1), & \text{if } l \neq \bar{l} \end{cases}, \tag{12}$$

The first Auto-Regressive (AR) model is adopted to represent the correlation feature of random acceleration, which is given by [34]:

$$\mathbf{b}_{k+1} = \alpha_\Phi \mathbf{b}_k + \boldsymbol{\omega}_k, \tag{13}$$

where $\alpha_\Phi \in (0, 1)$ is the reciprocal of the acceleration time constant. The random acceleration vector $\boldsymbol{\omega}_k = [\omega_x, \omega_y, \omega_z]_k^\intercal$ is a multivariate normal distribution with $\boldsymbol{\omega} \sim \mathcal{N}(0_{3\times1}, \sigma_\omega^2 \mathbf{I}_3)$, where $\mathbf{I}_3$ is an identity matrix of dimension 3×3. Hence, the

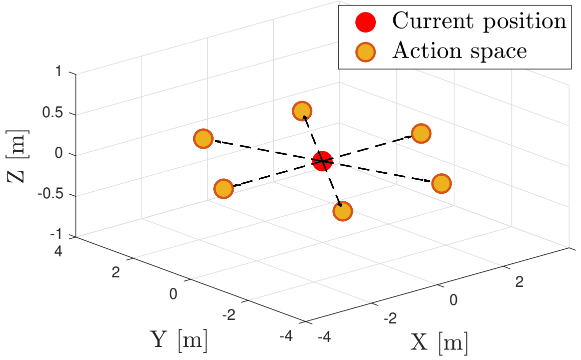


Fig. 2. Admissible control actions where UAV is at the origin and $N_\theta = 6$.

dynamics of the j-th FR at timestep k can be expressed by the following discrete-time system [35]:

$$\mathbf{x}_k^j = \mathbf{\Phi}\mathbf{x}_{k-1}^j + \mathbf{\Gamma}_\nu \boldsymbol{\nu}_k + \mathbf{\Gamma}_\omega \boldsymbol{\omega}_k. \tag{14}$$

where the matrices $\mathbf{\Phi}$, $\mathbf{\Gamma}_\nu$, and $\mathbf{\Gamma}_\omega$ are represented as follows:

$$\mathbf{\Phi} = \begin{bmatrix} \tilde{\mathbf{\Phi}} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \tilde{\mathbf{\Phi}} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & \tilde{\mathbf{\Phi}} \end{bmatrix}, \ \mathbf{\Gamma}_i = \begin{bmatrix} \tilde{\mathbf{\Gamma}}_i & 0_{3\times1} & 0_{3\times1} \\ 0_{3\times1} & \tilde{\mathbf{\Gamma}}_i & 0_{3\times1} \\ 0_{3\times1} & 0_{3\times1} & \tilde{\mathbf{\Gamma}}_i \end{bmatrix}, \tag{15}$$

$$\tilde{\mathbf{\Phi}} = \begin{bmatrix} 1 & \Delta k & \Delta k^2/2 \\ 0 & 1 & \Delta k \\ 0 & 0 & \alpha_\Phi \end{bmatrix}, \tilde{\mathbf{\Gamma}}_\nu = \begin{bmatrix} \Delta k^2/2 \\ \Delta k \\ 0 \end{bmatrix}, \tilde{\mathbf{\Gamma}}_\omega = \begin{bmatrix} \Delta k^2/2 \\ \Delta k \\ 1 \end{bmatrix}, \tag{16}$$

where the subscript i of the matrix $\tilde{\mathbf{\Gamma}}_i$ represents ν or ω, $0_{3\times1}$ is a zero matrix of dimension 3×1, and $0_{3\times3}$ is a zero matrix of dimension 3×3.

B. UAV Dynamics

We assume that a team of M UAVs operate in the environment and monitor the FRs. At timestep k, the state vector of the i-th UAV is represented by $\mathbf{u}^i = [\mathrm{u}_x^i, \mathrm{u}_y^i, \mathrm{u}_z^i]_k^\intercal$. The UAV dynamics are formulated as:

$$\mathbf{u}_k^i = \mathbf{u}_{k-1}^i + a_n = \mathbf{u}_{k-1}^i + \begin{bmatrix} d\cos(n\Delta_\theta) \\ d\sin(n\Delta_\theta) \\ 0 \end{bmatrix}, \tag{17}$$

where d is a constant distance that the UAVs can move at each timestep k and $\Delta_\theta = 2\pi/N_\theta$ is the unit steering angle. The action control a_n, $n = \{1, \ldots, N_\theta\}$ denotes the flight direction along x, y, and z axis. UAVs determine the flight direction by choosing one action from discrete action space $\{a_1, a_2, \ldots, a_{N_\theta}\} \in \mathcal{A}$.

C. UAV Sensing Model

We consider that each UAV is equipped with a range sensor that measures the distance between the i-th UAV and the j-th target. UAVs receive distance measurements from ground targets every timestep. The measurement model is represented as follows:

$$y_k^{ij} = h(\mathbf{c}_k^j, \mathbf{u}_k^i) + w_k^{ij} = \|\mathbf{c}_k^j - \mathbf{u}_k^i\|_2 + w_k^{ij}, \tag{18}$$

where $\mathbf{c}^j = [x^j, y^j, z^j]^\intercal$ is the j-th target position, the function $h\left(\mathbf{c}^j, \mathbf{u}^i\right)$ is euclidean distance of the i-th UAV, and the j-th target, and w_k^{ij} is measurement noise between the i-th UAV and the j-th target. Due to various obstacles in the environment, the UAVs receive LoS and NLoS measurements from targets as shown in Fig. 1. For this reason, we model the measurement noise w_k^{ij} as [18, 36]:

$$w_k^{ij} \sim \left[\lambda^{ij}\ \mathcal{N}(0, \sigma_{LoS}^2) + (1 - \lambda^{ij})\ \mathcal{N}(\mu_{NLoS}, \sigma_{NLoS}^2)\right], \tag{19}$$

where $\mathcal{N}(0, \sigma_{LoS}^2)$ denotes LoS measurement characteristics i.e., as a Gaussian distribution with zero mean and variance σ_{LoS}^2 and $\mathcal{N}(\mu_{NLoS}, \sigma_{NLoS}^2)$ denotes the NLoS measurements statistical profile i.e., as a Gaussian distribution with mean μ_{NLoS} and variance σ_{NLoS}^2. The measurement noise model is thus a mixture model of LoS and NLoS components, and the i-th UAV receives LoS component from the j-th target with probability λ^{ij}, which is formulated as follows:

$$\lambda^{ij} = p(\Theta^{ij}) = \frac{1}{1 + \alpha \exp\left[-\beta(\Theta^{ij} - \alpha)\right]} \tag{20}$$

where $\Theta^{ij} = \arcsin\left((\mathrm{u}_z^i - z^j)\ /\ \|\mathbf{c}^j - \mathbf{u}^i\|_2\right)$ is the elevation angle between the i-th UAV and the j-th target. The two parameters α and β, relate to the ratio of structured area to total land area and the number of buildings per unit land area [37].

V. Cramér-Rao lower bound of first responders state estimator

This section describes the CRLB of the FRs position, which is the main criterion to quantify the system performance. Also, we briefly introduce the optimal UAV joint control actions according to the CRLB as discussed in [36].

A. CRLB of FRs Position

CRLB is a lower bound of variance on the unbiased estimator, which represents achievable estimator performance. The CRLB of all target positions is formulated by:

$$\mathrm{var}(\hat{\mathbf{F}}_k) \geq \mathrm{tr}(\mathbf{J}^{-1}(\mathbf{F}_k)), \tag{21}$$

where $\mathrm{tr}(\cdot)$ means the trace of a square matrix, which is the sum of the diagonal elements of the matrix, and $\mathbf{F}_k = [\mathbf{c}_k^1, \ldots, \mathbf{c}_k^N]$ is a vector that contains the all target position. The Fisher information matrix (FIM) $\mathbf{J}(\mathbf{F}_k)$ is given by [38]:

$$\mathbf{J}(\mathbf{F}_k) = -E\left\{\frac{\partial^2 \ln \Lambda(\mathbf{Y}_k \mid \mathbf{X}_k)}{\partial \mathbf{F}_k^2}\right\}. \tag{22}$$

The joint measurement likelihood function considering the UAV sensing model presented in Subsection IV-C, is approximated by a single Gaussian distribution as follows:

$$\Lambda(\mathbf{Y}_k \mid \mathbf{X}_k) = \prod_{i=1}^{M}\prod_{j=1}^{N} \mathcal{N}(y_k^{ij} \mid h(\mathbf{c}_k^j, \mathbf{u}_k^i) + \mu_k^{i,j}, (\sigma_k^{ij})^2), \tag{23}$$

where $\mathbf{Y}_k = y_k^{ij}(i,j),\ i \in \{1,\dots,M\}, j \in \{1,\dots,N\}$ is distance measurement from the j-th target received by the i-th UAV at timestep k, and $\mathbf{X}_k(j) = \mathbf{x}_k^j,\ j \in \{1,\dots,N\}$ is the j-th target state at timestep k. According to the UAV sensing model, the mean and variance of the joint measurement are given by:

$$\mu_k^{ij} = (1-\lambda_k^{ij})\mu_{NLoS}, \tag{24}$$

$$\begin{aligned}(\sigma_k^{ij})^2 = \lambda_k^{ij}(\sigma_{LoS}^2 - (\mu_k^{ij})^2) \\ + (1-\lambda_k^{ij})(\sigma_{NLoS}^2 + \mu_{NLOS}^2 - (\mu_k^{ij})^2)\end{aligned} \tag{25}$$

The joint measurement likelihood function can be represented by the log-likelihood function $\ln \Lambda(\mathbf{Y}_k|\mathbf{X}_k)$:

$$\sum_{i=1}^{M}\sum_{j=1}^{N}\left\{\ln\frac{1}{\sqrt{2\pi}\sigma_k^{ij}} - \frac{(y_k^{ij} - h(\mathbf{c}_k^j, \mathbf{u}_k^i) - \mu_k^{ij})^2}{2(\sigma_k^{ij})^2}\right\}. \tag{26}$$

According to derivation in [36], the FIM $\mathbf{J}(\mathbf{F}_k)$ is represented in the form of a block diagonal matrix:

$$\mathbf{J}(\mathbf{F}_k) = \text{diag}([\mathbf{J}_1, \mathbf{J}_2, \dots, \mathbf{J}_N]), \tag{27}$$

The CRLB of all target state is expressed as the sum of each target's CRLB:

$$\text{var}(\hat{\mathbf{F}}_k) \geq \sum_{j=1}^{N} \text{tr}(\mathbf{J}_j^{-1}). \tag{28}$$

B. UAVs Optimal Control Actions

CRLB-based control is described as an extension to a greedy algorithm. At each timestep, candidate positions are determined by the UAV's current position and action space $\mathcal{A}$, defined in Subsection. IV-B. The CRLB for all candidate positions is calculated using the predicted target position $\tilde{\mathbf{x}}_k$ since the actual target position is unknown. UAVs select optimal action combination $\mathbf{U}_k^*$ corresponding to minimum CRLB among candidate positions, which is given by:

$$\mathbf{U}_k^* = \arg\min_{\mathbf{U}_k} \sum_{j=1}^{N} \text{tr}\left(\mathbf{J}_{j,\mathbf{U}_k}^{-1}\right), \tag{29}$$

where $\mathbf{U}_k = \{a_k^1, \dots, a_k^M\}$ is a combination of UAV control action, and $\mathbf{J}_{j,\mathbf{U}_k}$ represents FIM calculated by the j-th predicted target position and UAVs position changed by control action $\mathbf{U}_k$.

VI. The proposed deep reinforcement learning-based UAV trajectory control

We introduce a DRL-based multiple UAV control algorithm for accurate target state estimation. First, we present details of the DRL algorithm, including state, action, and reward design. Then, we describe the target state estimation using Bayesian filtering.

Figure 3 illustrates an overview of the DRL-based first responder tracking system. The system consists of two parts, FR estimation and DRL-based controller part. The FR estimation part is composed of time prediction stage and measurement update stage, where the states of FRs predicted and corrected according to the prediction density $p(\mathbf{x}_k|\mathbf{Y}_{1:k-1})$ and the posterior distribution $p(\mathbf{x}_k|\mathbf{Y}_{1:k})$, respectively. In the time prediction, the state of FRs $\tilde{\mathbf{x}}_k$ are predicted using a probabilistic model based on the target dynamics. After the time prediction stage, the DRL-based controller operates in order to adjust UAV's position. In the DRL-based UAV controller, the i-th UAV observes state $\mathbf{s}_k^i$, move its position $\mathbf{u}_k^i$ by taking action a_k^i, and gets a reward r_k^i through the UAV-environment interaction. The input of DRL-based controller (i.e., state of the UAV) is determined by the position of UAV and the predicted target position obtained in the time prediction of FR estimation part. Then, in the measurement update stage, the states of FRs $\hat{\mathbf{x}}_k$ are corrected through the measurement likelihood function obtained from distance measurements which are received by UAVs. Details are described in the next subsections.

A. DRL-based Controller Design

The components of the DRL-based UAV controller form a Markov Decision Process or MDP (i.e., Sec.III-A) and include the elements: state, action, reward function, and training process.

1) State

The state vector of the i-th agent at timestep k considers UAV position and target position, which is represented by $\mathbf{s}_k^i = [\mathbf{u}_k^i, \mathbf{p}_k^{i1}, \dots, \mathbf{p}_k^{iM}, \mathbf{q}_k^{i1}, \dots, \mathbf{q}_k^{iN}] \in \mathbb{R}^{3(M+N)}$. The state vector of the i-th agent consists of three parts:

- $\mathbf{u}^i$: absolute coordinates of the i-th agent.
- $\mathbf{p}^{i\bar{i}}$: relative coordinates between the i-th agent and the $\bar{i}$-th agent.
- $\mathbf{q}^{ij}$: relative coordinates between the i-th agent and the j-th target.

The relative coordinates between the i-th agent and the $\bar{i}$-th agent is given by $\mathbf{p}^{i\bar{i}} = (\mathbf{u}^{\bar{i}} - \mathbf{u}^i)^\intercal$, where $i, \bar{i} \in \{1,\dots,M\}$, and $i \neq \bar{i}$. Each relative coordinates $\mathbf{p}^{i\bar{i}}$ is concatenated into one vector $\left[\mathbf{p}^{i1}, \dots \mathbf{p}^{iM}\right] \in \mathbb{R}^{1\times 3(M-1)}$. The relative coordinates between the i-th agent and the j-th target is given by $\mathbf{q}^{ij} = (\tilde{\mathbf{c}}^j - \mathbf{u}^i)^\intercal$, where $j \in \{1,\dots N\}$ and $\tilde{\mathbf{c}}_k^j$ is the predicted target position extracted from the predicted target state $\tilde{\mathbf{x}}_k^j$. Each relative coordinates $\mathbf{q}^{ij}$ is concatenated into one vector in order $\left[\mathbf{q}^{i1}, \dots, \mathbf{q}^{iN}\right] \in \mathbb{R}^{1\times 3N}$. The input size of DRL varies with the number of agents and targets.

2) Action

In each timestep, the i-th agent selects action a_k^i from discrete action space $\mathcal{A}$ defined in Sec. IV-B. The selected action determines the next position of the UAV. If the action combination selected by the DRL-based controller is likely to cause collisions between UAVs, the selected action is replaced by another action among the action space.

3) Reward

The agent observes its state from the environment and takes action. Through this interaction, each agent receives a scalar reward from the environment. UAVs make a decision to maximize cumulative reward, and they aim to improve target tracking performance by minimizing CRLB Ψ. The reward of

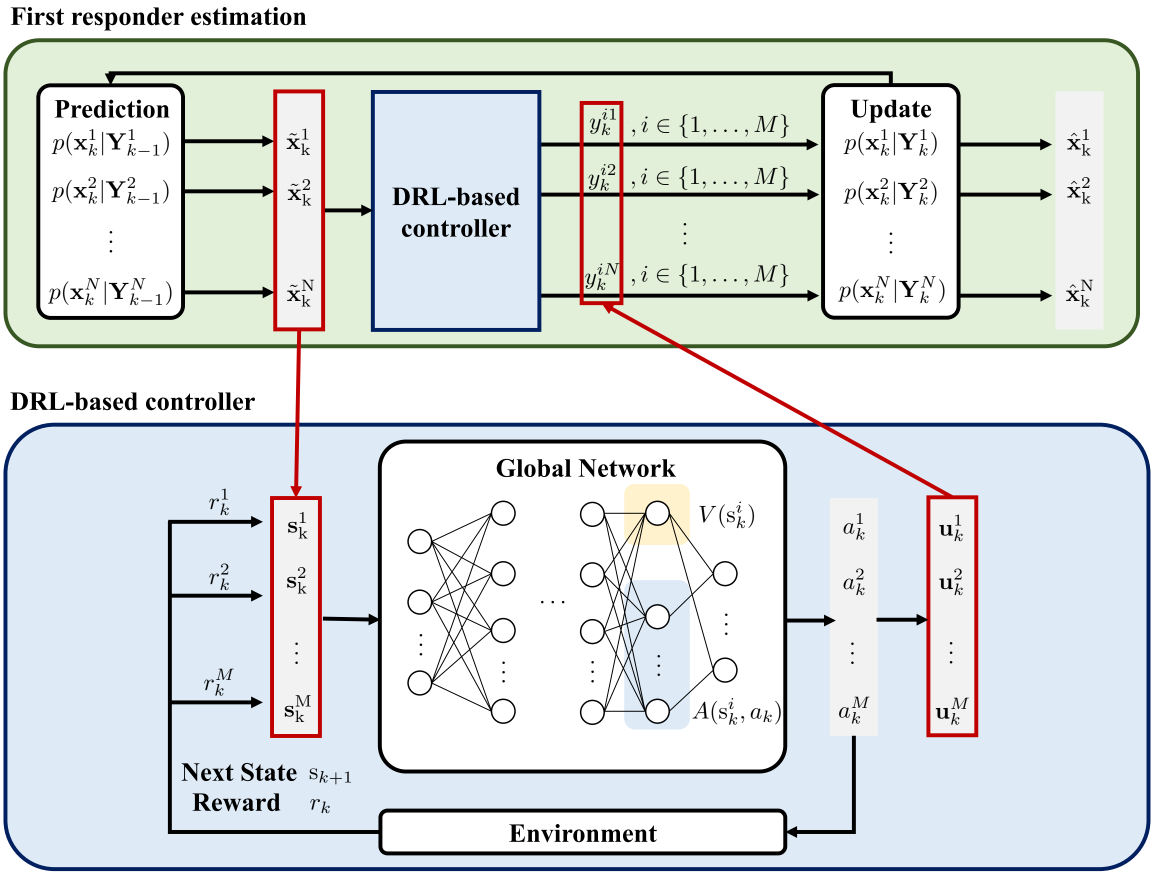


Fig. 3. An overview of the DRL-based first responder tracking system. The green part depicts that the first responder's state is updated by prediction density in the time prediction stage, and the posterior distribution in the measurement update stage. The blue part represents DRL-based control. Agents observe their state from the environment and select actions according to the dueling network where the yellow part is the value function estimator and the blue part is the advantage estimator. For agent-environment interaction, Each agent learns a policy to maximize Q-value and share a global Q-value estimator.

the i-th agent at timestep k comprises of three sub-rewards, and it is formulated as follows:

$$r_k^i = R_{1,k} + R_{2,k} - R_{3,k}^i. \tag{30}$$

The reward r_k^i aggregates sub-rewards into a single scalar, whose form is most widely used in multi-objective problems [39]. The selected action combination from the DRL-based control should be a solution, called Pareto-optimal, which maximizes the reward in a multi-objective problem [40]. In above equation, R_1 and R_2 correspond to global reward, and R_3 to difference reward [41, 42]. The *global reward*, denoted as $G(\mathbf{s}_k, a_k)$, is given to the agents based on the utility of the entire system. All agents receive the same global reward, regardless of the effect of each agent's action on the entire system. On the other hand, *difference reward* D_i quantifies each agent's individual contribution to the entire system. The difference reward is given by:

$$D_i(\mathbf{s}_k^i, a_k^i) = G(\mathbf{s}_k, a_k) - G(\mathbf{s}_k^{-i}, a_k^{-i}) \tag{31}$$

where the counterfactual $G(\mathbf{s}_k^{-i}, a_k^{-i})$ is the global reward without the i-th agent's contribution to the system, calculated by assuming the i-th agent is not present. As mentioned before, global reward functions $R_{1,k}$ and $R_{2,k}$ are given by $G_1(\mathbf{s}_k, a_k)$ and $G_s(\mathbf{s}_k, a_k)$, respectively. For $R_{3,k}^i$, the utility of the entire system is represented by $G_3(\mathbf{s}_k, a_k)$, and the difference reward of the i-th agent $D_{3,k}^i$ is expressed by the difference between $G_3(\mathbf{s}_k, a_k)$ and $G_3(\mathbf{s}_k^{-i}, a_k^{-i})$.

The **CRLB variation reward**, $R_{1,k}$, is a global reward related to how much CRLB is decreased:

$$R_{1,k} = \eta_1 \times G_1(\mathbf{s}_k, a_k) = \eta_1 \times \left(\frac{\Delta_\Psi - \Delta_m}{\Delta_M - \Delta_m} + \kappa_1 \right). \tag{32}$$

The utility of the entire system is the CRLB difference $\Delta_\Psi = \Psi_{k-1} - \Psi_k$ between at timestep $k-1$ and timestep k. All agent get positive R_1 when they move to a position with lower CRLB. The absolute value of the CRLB difference is substantial when the tracking performance significantly improves or worsens, comparing timestep k and timestep $k-1$. The tuning parameters Δ_m and Δ_M determined via experiments are the minimum bound and the maximum bound of CRLB difference, respectively, where the 95% of CRLB difference is in the range of $[\Delta_m, \Delta_M]$. Two parameters are used to scale the CRLB difference and make the 95% of the first term in parentheses in the range of [0,1]. The parameter κ_1 adjusts the range of $G_1(\mathbf{s}_k, a_k)$ to $[-0.5, 0.5]$. The parameter η_1 denotes the magnitude of reward functions R_1.

The **CRLB magnitude reward**, $R_{2,k}$, is a global reward representing how small the CRLB Ψ_k is at time k:

$$R_{2,k} = \eta_2 \times G_2(\mathbf{s}_k, a_k) = \eta_2 \times \left(e^{-\delta \cdot \Psi_k} + \kappa_2\right). \tag{33}$$

For reward function R_2, the utility is CRLB of the target state estimator. As CRLB increases, the rewards awarded to all agents decrease exponentially. The tuning parameter δ determined by experiments is the degree to which the reward is reduced. The parameter κ_2 adjusts the range of $G_2(\mathbf{s}_k, a_k)$ to $[-0.5, 0.5]$ and the value in parentheses is positive if CRLB Ψ_k is smaller than $0.7 * (1/\delta)$. The parameter η_2 changes the magnitude of reward functions R_2.

The **difference reward**, $R^i_{3,k}$, quantifies the i-th agent's contribution to tracking performance:

$$R^i_{3,k} = \eta_3 \times D^i_{3,k}, \tag{34}$$

where the tuning parameter η_3 changes the range of the reward. We define the reward $D^i_{3,k}$ as follows:

$$D^i_{3,k} = \frac{G_3(\mathbf{s}_k, a_k) - G_3(\mathbf{s}^{-i}_k, a^{-i}_k)}{(\mathbf{J}^{-1}_{j^\star})_{-i}} = \frac{\mathbf{J}^{-1}_{j^\star}}{(\mathbf{J}^{-1}_{j^\star})_{-i}} - 1, \tag{35}$$

with

$$j^\star = \underset{j \in \{1,2\ldots N\}}{\text{argmax}} \mid \mathbf{J}^{-1}_j - (\mathbf{J}^{-1}_j)_{-i} \mid, \tag{36}$$

where $\mathbf{J}^{-1}_j$ is inverse FIM of the j-th target for all agents in the system, and $(\mathbf{J}^{-1}_j)_{-i}$ is inverse FIM of the j-th target when the i-th agent is excluded from the entire system. The notation $j^\star$ means a target with the largest difference between $\mathbf{J}^{-1}_j$ and $(\mathbf{J}^{-1}_j)_{-i}$. The utility of $R^i_{3,k}$ considers CRLB of the $j^\star$-th target; in other words, the effect of the presence of the i-th agent on the $j^\star$-th target tracking performance. The utility of UAV system $G_3(\mathbf{s}_k, a_k) = \mathbf{J}^{-1}_{j^\star}$ is the CRLB of the $j^\star$-th target state estimator, and $G_3(\mathbf{s}^{-i}_k, a^{-i}_k) = (\mathbf{J}^{-1}_{j^\star})_{-i}$ is the CRLB of the $j^\star$-th target state estimator when the i-th agent is absent. The difference between $G_3(\mathbf{s}_k, a_k)$ and $G_3(\mathbf{s}^{-i}_k, a^{-i}_k)$ is normalized to $(\mathbf{J}^{-1}_j)_{-i}$ to adjust the range of $D^i_{3,k}$ to $[-1, 0]$. The difference reward of the i-th agent approaches -1 when the i-th agent has a significant impact on the CRLB of the $j^\star$-th target. Therefore, the total reward r^i_k is obtained by subtracting the difference reward $R^i_{3,k}$.

4) Reinforcement Learning Training

A conventional DRL process is introduced in Subsection III-B. The blue part in Fig. 3 illustrates the overall structure of the DRL-based multiple UAV controller. Each agent observes its state and selects an action from the global dueling network described in Subsection III-C. The global network is updated by the interaction of distributed agents. Details of the multiple UAVs control algorithm are provided in Alg. 1. In the beginning, the Q network is initialized with random weights of θ. The weights of target Q network are replicated as the weights of Q network $\theta^- = \theta$. Also, a replay memory, which stores the recent N_D transition tuples, is initialized (Line 1-3). With every new episode, the DRL-environment is initialized, and thus the agents learn various tracking strategies by trial and error during each episode composed of consecutive K timesteps without any stopping criterion (Line 4-6). Each agent selects an action corresponding to maximum Q value with probability $1 - \epsilon$ or randomly with probability ϵ. The probability ϵ decreases as the training is repeated (Line 7-8). The transition sample $\left(\mathbf{s}^i_k, a^i_k, r^i_k, \mathbf{s}^i_{k+1}\right)$ is stored in the replay memory $\mathcal{D}$ (Line 9-10). A minibatch consisting of N_B transition tuples is sampled uniformly from all transitions in replay memory $\mathcal{D}$ to calculate the target values and loss function. The weights of the Q network are updated, while reducing the loss function with the optimizer. Then, target Q network is updated, duplicating the weights of Q network every $\mathcal{N}$ timesteps (Lines 12-16).

Algorithm 1: Multi-UAV control for target tracking based on DRL

```
   Input  : State vector of $i$-th agent
   Output: Action of agent $i$-th agent
1  Initialize Q network $Q$ with random weights of $\theta$ ;
2  Initialize target Q network $Q^-$ with weights $\theta^- = \theta$ ;
3  Initialize replay memory $\mathcal{D}$ to capacity $N_D$;
4  for Episode $:= 1, \ldots, N_t$ do
5      Initialize environment ;
6      for Timestep $k := 1, \ldots, K$ do
7          for Agent $i := 1, \ldots, M$ do
8              Select a greedy action
               $a^i_k = \arg\max_{a_k} Q^i(\mathbf{s}^i_k, a_k)$ with probability
               $1 - \epsilon$ or a random action with probability
               $\epsilon$ ;
9              Execute action $a^i_k$ ;
10             Store the transition sample
               $\left(\mathbf{s}^i_k, a^i_k, r^i_k, \mathbf{s}^i_{k+1}\right)$ in replay memory ;
11         end
12         Sample minibatch from replay memory $\mathcal{D}$;
13         Calculate target value using (7);
14         Calculate loss value using (6);
15         Update Q network;
16         Update target Q network every $\mathcal{N}$ time steps;
17     end
18 end
```

B. FRs State Estimation

We use particle filtering [43] to estimate the posterior distribution of the target states given the noisy measurements. As shown in Fig. 3, the target position is first predicted, and then used as the input to the DRL-based controller. After the locations of UAVs are determined by the DRL-based controller, the target position is corrected in the measurement update step. The main concept of the particle filter is to construct a posterior distribution $p(\mathbf{x}_k|\mathbf{Y}_{1:k})$ of target state $\mathbf{x}_k$, given measurements $\mathbf{Y}_{1:k} = \{y_1, y_2, \ldots, y_k\}$ up to timestep k. The time prediction and measurement update is computed as follows:

$$p(\mathbf{x}_k|\mathbf{Y}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{Y}_{1:k-1}) d\mathbf{x}_{k-1}, \tag{37}$$

$$p(\mathbf{x}_k|\mathbf{Y}_{1:k}) = \frac{p(y_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Y}_{1:k-1})}{\int p(y_k|\mathbf{x}_k), p(\mathbf{x}_k|\mathbf{Y}_{1:k-1})d\mathbf{x}_k} \tag{38}$$

where $p(\mathbf{x}_k|\mathbf{x}_{k-1}) \sim \mathcal{N}(\mathbf{\Phi}\mathbf{x}_{k-1} + \mathbf{\Gamma}_\nu \boldsymbol{\nu}_k, \mathbf{\Gamma}_\omega \mathbf{\Gamma}_\omega^\intercal \sigma_\omega)$ is a probabilistic model of the state evolution (transitional density) defined by (14), and $p(y_k|\mathbf{x}_k)$ is a measurement likelihood function defined by (18).

The green part in Fig. 3 shows the target state estimation process in the proposed system. In the time prediction stage, the j-th predicted target state $\tilde{\mathbf{x}}_k^j$ is determined by prediction density $p(\mathbf{x}_k^j|\mathbf{Y}_{1:t-1}^j)$ and is calculated as follows:

$$\tilde{\mathbf{x}}_k^j = \int \mathbf{x}_k^j \; p(\mathbf{x}_k^j|\mathbf{Y}_{1:k-1}^j)d\mathbf{x}_k^j. \tag{39}$$

The predicted target state $\tilde{\mathbf{x}}_k^j$ is used as input of the DRL-based controller to select the UAVs' actions. Then each UAV moves to their new position according to actions selected by the DRL-based controller and receives distance measurements from the ground targets.

The posterior distribution $p(\mathbf{x}_k|\mathbf{Y}_{1:k})$ depends on measurement likelihood $p(y_k|\mathbf{x}_k)$ calculated by distance measurements. The measurement likelihood function of the j-th target is given by:

$$p(y_k^{1j}, \ldots, y_k^{Mj}|\mathbf{x}_k^j, \mathbf{u}_k^1, \ldots, \mathbf{u}_k^M) = \prod_{i=1}^{M} p(y_k^{ij}|\mathbf{x}_k^j, \mathbf{u}_k^i), \tag{40}$$

where $p(y_k^{ij}|\mathbf{x}_k^j, \mathbf{u}_k^i) = \mathcal{N}\Big(y_k^{ij}|h(\tilde{\mathbf{c}}_k^j, \mathbf{u}_k^i) + \mu_k^{ij}, (\sigma_k^{ij})^2\Big)$ is measurement likelihood function of the j-th target and the i-th UAV. The posterior distribution is obtained by Bayes' theorem (38) and the estimated target position $\hat{\mathbf{x}}_k^j$ is obtained as follows:

$$\hat{\mathbf{x}}_k^j = \int \mathbf{x}_k^j \; p(\mathbf{x}_k^j|\mathbf{Y}_{1:k}^j)d\mathbf{x}_k^j. \tag{41}$$

VII. Simulation Results and Analysis

To evaluate the performance of the proposed approach, we have divided our evaluation into five main parts. The first subsection introduces settings for target tracking simulation. In the following two subsections, the results of single-target tracking and multi-target tracking are presented. We present the CRLB of the target state estimator and localization error to verify the tracking performance of DRL-based control, comparing to CRLB-based control [36] (mentioned in Section V-B), Genetic Algorithm (GA)-based control [44] and Discrete Particle Swarm Optimization (DPSO)-based control [45]. The DRL-based control should maintain the CRLB of the target estimator during the entire timesteps at the same level as the CRLB-based control, where it always selects an optimal action combination. The localization error is defined as Mean Squared Error (MSE) between the estimated target position and real target position, which is formulated as:

$$\text{MSE} = \mathbb{E}\Bigg[\sum_{j=1}^{N} (\mathbf{x}^j - \hat{\mathbf{x}}^j)^2\Bigg] \tag{42}$$

TABLE I
Simulation Parameters

Parameters	Unit	Value	
		Experiment 1	Experiment 2
N		1	2
T_d	[sec]	1	1
α_Φ		0.95	0.9
σ_ω^2	$[\text{m/s}^2]^2$	0.5^2	0.5^2
p_l	$[\text{m/s}^2]$	0.1	0.1
M		3	4
N_θ		4	4
d	[m]	2.5	3
Δ_m		-20	-20
Δ_M		20	20
κ_1		-0.5	-0.5
κ_2		0.5	0.5
δ		0.02	0.02
σ_{LoS}	[m]	0.5	0.8
μ_{NLoS}	[m]	5	5
σ_{NLoS}	[m]	5	5
α		0.7	0.5
β		10	10
η_1		20	10
η_2		10	20
η_3		10	10
K		100	100
Learning rate		0.0001	0.0001
Training iteration N_t		10000	20000
Population (DPSO)		5	5
Max. generation (DPSO)		10	10
Population (GA)		50	50
Max. generation (GA)		50	100

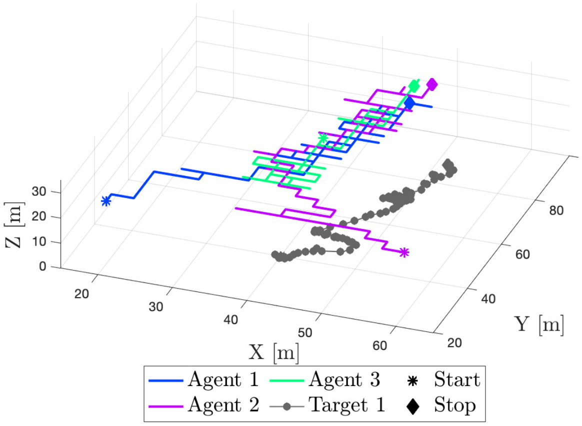


Fig. 4. Trajectories of three UAVs and one target.

In the fourth subsection, we present two metrics, reward and run-time, to prove that the proposed approach is effective for real-time tracking. Finally, the last subsection shows that the CRLB-based control needs to be replaced with alternative control methods in larger-scale problems.

A. Simulation Setup

Our experiments are conducted on Ubuntu 16.04 server with Intel i7-4790K. We use three fully-connected layers with 50 neurons and ReLU activation, $\text{Relu}(x) = \max(0, x)$. The third layer is connected to two streams, estimators of value function and advantage function. The size of replay memory is $N_D =$

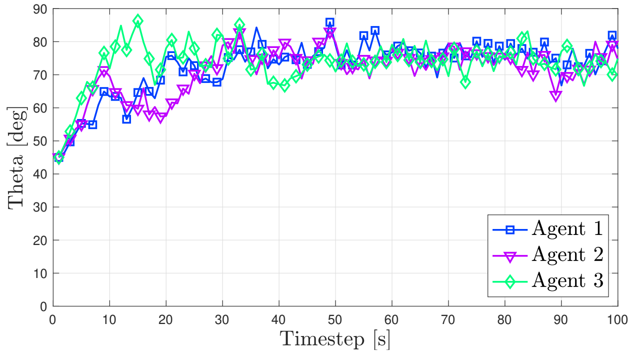


Fig. 5. Elevation angle between UAVs and target.

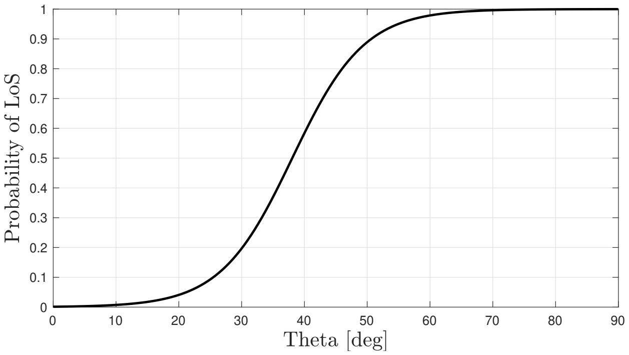


Fig. 6. Probability of LoS ($\alpha = 0.7$, $\beta = 10$) in urban environment.

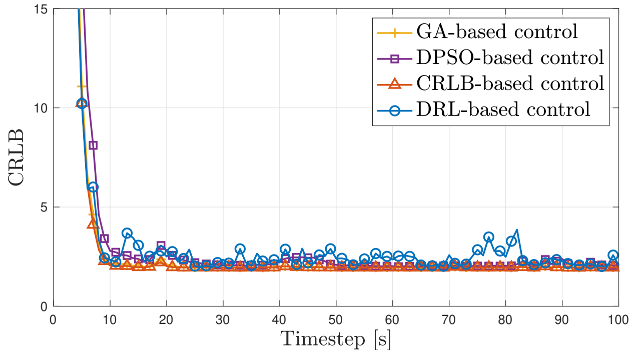


Fig. 7. CRLB of four UAV control methods.

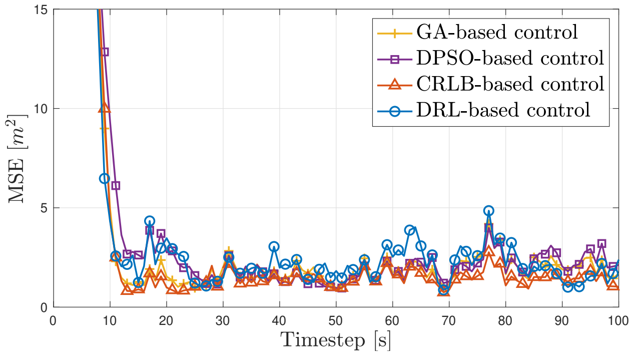


Fig. 8. MSE of four UAV control methods over 100 Monte Carlo experiments.

5000, and minibatch is randomly sampled consisting of $N_B = 128$ transition tuples selected from replay memory $\mathcal{D}$. After the training stage, Each agent selects its action corresponding to the maximum Q-value from the trained network to verify the performance of the trained network. Detailed simulation parameters are presented in Table I.

B. Experiment 1 : Single Target Tracking

In this experiment, there is one ground target, whose initial state vector is set to $[x, \dot{x}, \ddot{x}, y, \dot{y}, \ddot{y}, z, \dot{z}, \ddot{z}]^\intercal = [40, 0.4, 0, 40, 0.4, 0, 0, 0, 0]^\intercal$ with units m, m/s, m/s^2, m, m/s, m/s^2, m, m/s and m/s^2. Initial positions of three UAVs are $[x, y, z] = [20, 25, 25]^\intercal$m, $[60, 25, 25]^\intercal$m and $[40, 65, 25]^\intercal$m. The discrete acceleration level is set to $\mathbb{L} = \mathcal{L}_x \times \mathcal{L}_y \times \mathcal{L}_z = \{(0, 0, 0), (1, 0, 0), (-1, 0, 0), (0, 1, 0), (0, -1, 0)\}$ in units of m/s^2.

Fig. 4 shows 3D trajectories of three UAVs, and Fig. 5 presents the elevation angle between UAVs and the target. Fig. 6 is the probability of LoS where the environment is urban. At the initial state, three UAVs are the same distance away from the target. They maintain elevation angles about 45° with the target, and the probability that three UAVs receive LoS is 0.75. When UAVs start monitoring missions, all UAVs the track ground target for entire timestep. At timestep $1 \leq k \leq 10$, Three UAVs move closer to the target position around $[40, 40, 0]m$. The elevation angle increases from 45° to over 65°, and UAVs receive get LoS measurement from target with over probability of 0.99. Between timestep $10 \leq k \leq 50$, the target begins to move long distances, as its speed increases. UAVs adjust their flight direction to catch up with the target. When timestep is between 50 and 80, the target hovers around [50m, 70m, 0m], and UAVs also fly close to the target. Elevation angles of three UAVs are around 75° during this time, and they receive LoS measurement with probability of 0.998. After timestep $k \geq 80$, UAVs select their action that move to trajectory of the target while keeping a certain distance from the target. Through this results, we confirmed that each UAVs select their own actions according to the trained network, and UAVs adjust their trajectories where they receive LoS measurement from target with a high probability.

Fig. 7 and Fig. 8 present CRLB and localization error when UAVs track the target moving in the trajectory shown in Fig. 4 by four UAV controls: DRL-based control, CRLB-based control, DPSO-based control, and GA-based control. The CRLB and localization error of each control are averaged over 100 Monte Carlo experiments. The initial CRLB is around 178.78 for four control schemes. During the entire timestep, CRLB and MSE of the CRLB-based control decrease more than the other controls because UAVs adjust their position corresponding to minimum CRLB among candidate positions. After timestep $k \geq 10$, CRLB and MSE of all control schemes decrease to about 2.2, although there are fluctuations in CRLB and MSE of four control schemes. The CRLB and tracking error depict that the three control methods excluding the CRLB-based control have a level of CRLB that is not significantly different from the CRLB of the CRLB-based control. Through this simulation results, we observe that DRL-

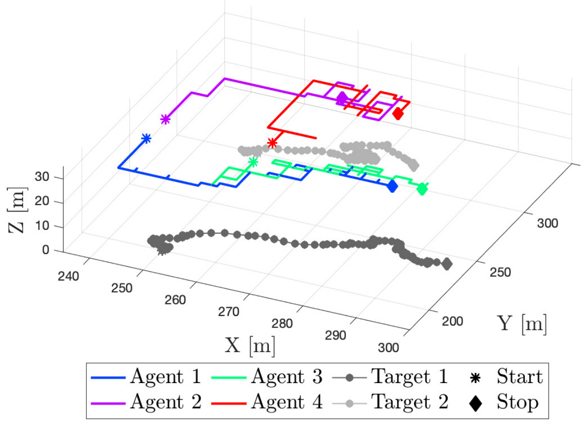


Fig. 9. Trajectories of four UAVs and two targets.

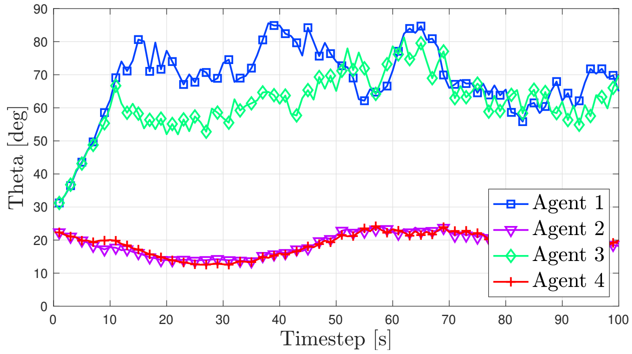


Fig. 10. Elevation angle between four agents and target 1.

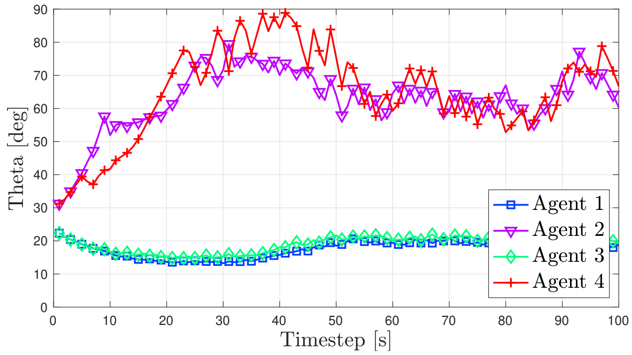


Fig. 11. Elevation angle between four agents and target 2.

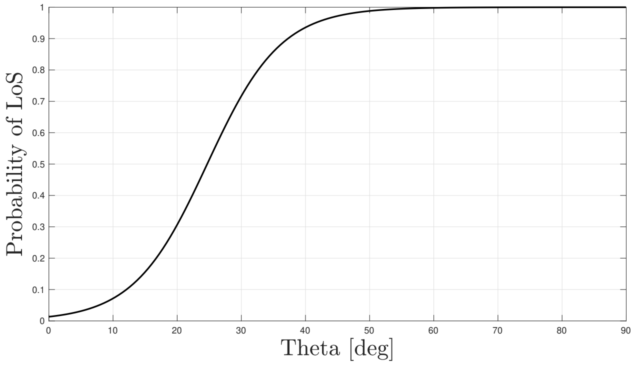


Fig. 12. Probability of LoS ($\alpha = 0.5$, $\beta = 10$) in rural environment.

based control achieves comparable tracking performance to the CRLB-based control which is the optimal control scheme.

C. Experiment 2 : Multiple Target Tracking

In this experiment, there are two targets whose initial vector are $[x, \dot{x}, \ddot{x}, y, \dot{y}, \ddot{y}, z, \dot{z}, \ddot{z}]^\intercal = [250, 0, 0, 200, 0, 0, 0, 0, 0]$ and $[250, 0, 0, 300, 0, 0, 0, 0, 0]$ with units m, m/s, m/s^2, m, m/s, m/s^2, m, m/s, and m/s^2. Four UAVs are placed in a square shape in the middle of two targets. The initial state vectors of four UAVs are $[x, y, z] = [240, 240, 25]^\intercal\text{m}$, $[240, 260, 25]^\intercal\text{m}$, $[260, 240, 25]^\intercal\text{m}$ and $[260, 260, 25]^\intercal\text{m}$. The discrete acceleration level is $\mathbb{L} = \mathcal{L}_x \times \mathcal{L}_y \times \mathcal{L}_z = \{(0, 0, 0),\ (1, 0, 0),\ (-1, 0, 0),\ (0, 1, 0),\ (0, -1, 0)\}$ in units of m/s^2.

Fig. 9 is 3D trajectories of four UAVs and two targets for entire timesteps. Fig. 10 and Fig. 11 are elevation angles between four UAVs and each target. Fig. 12 shows the probability that UAVs receive LoS measurement in the rural environment. In this environment, it is possible to ensure sufficient LoS at a lower elevation angle than the environment in Experiment 1. When all UAVs and targets are in the initial state $k = 1$, agent 1 and agent 3 are close to target 1 with the same distance. The other two agents are far from target 1, but closer to target 2. Four agents maintain elevation angles of 30° with the target, which is closer to themselves. It means that they receive LoS component with a probability of 0.7 from the closer target. For target which is far from all UAVs, the initial elevation angle is about 20° and UAVs obtain LoS component with a probability of 0.3. For timesteps $1 \le k \le 10$, agent 1 and agent 3 move in $-y$ direction to get closer to target 1. They achieve an elevation angle at least 50° with target 1, and ensure LoS measurement with a probability of 0.98. Likewise, agent 2 and agent 4 start to move toward the target 2 in the $+y$ direction. Two agents maintain elevation angle over 40°, collecting LoS measurement with a probability of 0.93. During timestep $10 \le k \le 30$, agent 1 and agent 3 move toward the target 1, and agent 2 and agent 4 track target 2. Two agents are assigned to one target and ensure sufficient LoS measurement from the assigned target at least a probability of 0.93. The two targets have different trajectories between the timestep $30 \le k \le 55$, which means the target 1 moves long-distance about from [250m, 215m, 0m] to [280m, 245m, 0m] and the target 1 goes around [265m, 315m, 0m]. To obtain LoS measurement from each assigned target, agent 1 and agent 3 move similar to the trajectories of target 1; in the same manner, agent 2 and agent 4 fly in the vicinity of target 2. For that period, each agent achieves the elevation angle over 50° with the assigned target, and receives LoS measurement with a probability of more than 0.98. After timestep $k \ge 55$, all UAVs steer their flight direction properly, maintaining the elevation angle with two targets over 50° to guarantee sufficient LoS measurements. It is confirmed that the action combination selected from the DRL-based controller enables the group of UAVs to maintain a high elevation angle and collect LoS

TABLE II
REWARD OF DRL-BASED CONTROL AND CRLB-BASED CONTROL IN EVALUATION STAGE

Experiment		Reward		
		R_{diff}	R_{glob}	R_{tot}
Exp. 1	CRLB-based control	29.95	5.22	45.64
	DRL-based control	14.22	3.55	24.67
Exp. 2	CRLB-based control	39.99	12.72	90.91
	DRL-based control	36.93	9.19	73.70

measurement from the multiple targets.

Fig. 13 and Fig. 14 illustrate CRLB and localization error, where the results are average over 100 Monte Carlo experiments. For four control schemes, it is shown that CRLB is greatly reduced from 1123 to 10 between timestep $1 \leq k \leq 10$. During that time, CRLB-based control has the best performance among the other controls because CRLB-based control enables UAVs to select action combinations corresponding to minimum CRLB. For timestep $10 \leq k \leq 80$, the three control methods except for the GA-based control maintain CRLB values between timestep 3 and 8; however, the GA-based control has a noticeably large CRLB among the all UAV control schemes. After $k \geq 80$, the tracking performance of GA-based control improved, and its CRLB value is reduced to 8; hence, all UAV controls maintain similar performance. For the DRL-based control, it maintains the low CRLB value of about 5 and attains similar tracking performance as the CRLB-based control for the entire timestep. Besides, as shown in Fig. 14, localization error follows the same tendency as CRLB. We observe that DRL-based control achieves the comparable tracking performance to the CRLB-based control and performs well in the multiple target tracking scenarios.

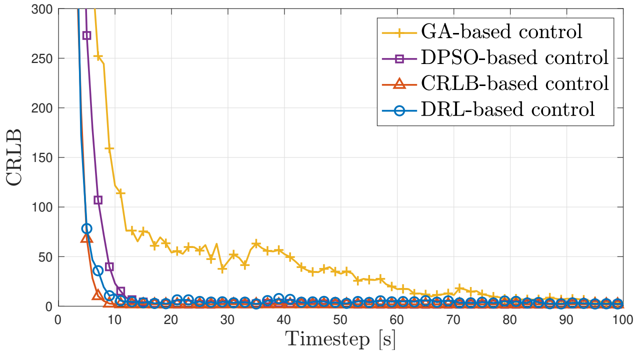


Fig. 13. CRLB of four UAV control methods

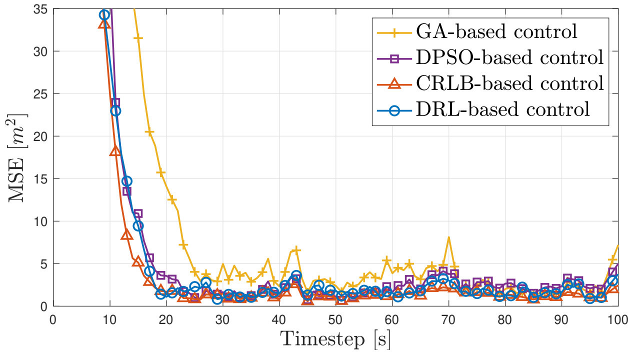


Fig. 14. MSE of four UAV control methods over 100 Monte Carlo experiments.

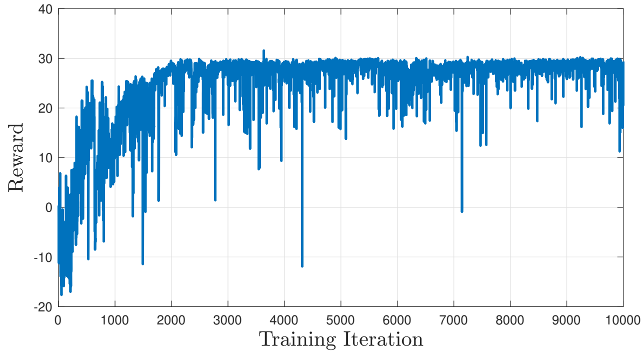


Fig. 15. Total reward curve versus the training iteration of Experiment 1.

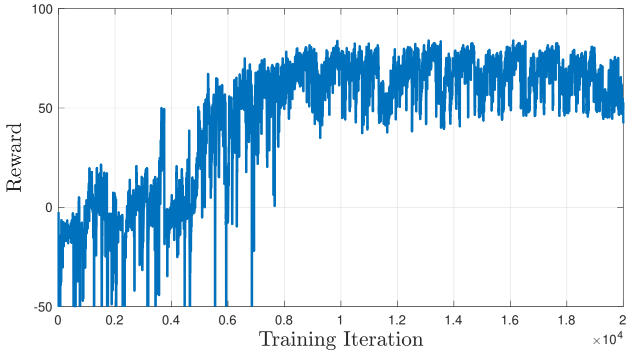


Fig. 16. Total reward curve versus the training iteration of Experiment 2.

D. Reinforcement Learning Performance

We investigate the performance of DRL-based control concerning the reward function that all agents receive during the training stage and evaluation stage first. The cumulative difference reward R_{diff}, cumulative global reward R_{glob} and cumulative total reward R_{tot} in one iteration are calculated as follows:

$$R_{diff} = \frac{1}{K} \sum_{k=1}^{K} \sum_{i=1}^{N} R_{3,k}^{i}, \tag{43}$$

$$R_{glob} = \frac{1}{K} \sum_{k=1}^{K} \left(R_{1,k} + R_{2,k} \right), \tag{44}$$

$$R_{tot} = \frac{1}{K} \sum_{k=1}^{K} \sum_{i=1}^{N} r_{k}^{i}, \tag{45}$$

where K is total timestep, presented in Table I. Fig. 15 and Fig. 16 illustrate the cumulative total reward R_{tot} received by all agents for training iteration in Experiment 1 and Experiment 2, respectively. In Fig. 15 and Fig. 16, total rewards increase and converge over the whole training stage, and it means that agents learn policy to maximize the reward by repeating the training iteration. Table II represents R_{diff}, R_{glob} and R_{tot} of two control methods in the evaluation

TABLE III
COMPARISON OF RUN-TIME BY CONTROL METHODS

Control schemes	Run-time $[s]$	
	Experiment 1	Experiment 2
GA-based control	0.74	2.02
DPSO-based control	0.47	1.18
CRLB-based control	0.56	1.21
DRL-based control	**0.01**	**0.02**

stage. In CRLB-based control, UAVs receive greater R_{diff}, R_{glob} and R_{tot} than DRL-based control because UAVs always take optimal action with minimum CRLB at their current positions. There is noticeable difference in R_{diff} between two controls, but DRL-based control achieves comparable R_{glob} to CRLB-based control in the evaluation stage. It means that although the contribution of individual UAVs to the tracking system is low, DRL-based control maintains similar tracking performance to CRLB-based control by building adequate geometry for target tracking.

There is a distinct difference between DRL-based control and three control methods in run-time. Run-time increases with the number of UAVs, the number of targets, and the size of action space. First, the run-time complexity of the CRLB-based control is $O(N_\theta^M)$, which increases with the size of action space N_θ to the power of the number of UAVs M because the CRLB-based controller calculates the CRLB of possible action combinations to determine optimal action combination corresponding to the minimum CRLB. The run-time complexity of the DRL-based control is $O(1)$ because the UAVs select their actions from the trained network. In order to verify the suitability of DRL-based control in real-time tracking, we present the time it takes for all UAVs to select their actions every timestep, as shown in Table III. In experiment 1, DRL-based control takes 0.01 seconds to select one action; however, CRLB-based control takes 56 times longer than DRL-based control, and the run-time of DPSO-based control and GA-based control requires about 47 times and 77 times than that of DRL-based control, respectively. In experiment 2, DRL-based control spends 0.02 seconds taking one action. DRL-based control is about 61 times faster than CRLB-based control, 59 times faster than DPSO-based control, and 101 times faster than GA-based control. It is confirmed that DRL-based control achieves comparable performance to CRLB-based control and is more suitable for real-time tracking than the existing algorithms.

E. Performance in Larger-scale Problem

The CRLB-based control is intractable in the larger-scale problem with numerous UAVs and large size of action space because the complexity of the CRLB-based control to select one action combination increases exponentially with the number of UAVs. Thus, an alternative control method is needed such as DRL-based control, DPSO-based control, and GA-based control. Since the previous two experiments are relatively small-scale problems, the advantages of the alternative control method are not noticeable. The GA-based control takes more time than the CRLB-based control in Experiment 1 and Experiment 2. However, in larger-scale problems, the DPSO-based control and GA-based control takes less time than the CRLB-based control because the DPSO-based control and GA-based control find a sub-optimal solution by iteratively improving the candidate solutions.

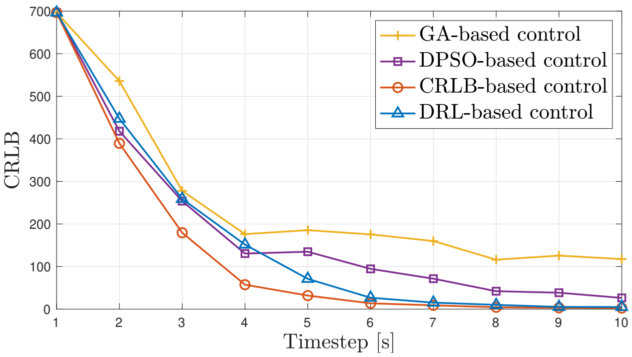


Fig. 17. CRLB of four control methods in the larger-scale problem.

TABLE IV
COMPARISON OF RUN-TIME BY CONTROL METHODS
IN LARGER-SCALE PROBLEMS

Control schemes	Run-time $[s]$	
	Mean	Std
GA-based control	19.01	0.22
DPSO-based control	18.92	0.14
CRLB-based control	28.94	0.98
DRL-based control	0.02	0

In this experiment, there are one target whose initial vector are $[x, \dot{x}, \ddot{x}, y, \dot{y}, \ddot{y}, z, \dot{z}, \ddot{z}]^\intercal = [250, 0, 0, 200, 0, 0, 0, 0, 0]$ with units m, $\mathrm{m/s}$, $\mathrm{m/s^2}$, m, $\mathrm{m/s}$, $\mathrm{m/s^2}$, m, $\mathrm{m/s}$, and $\mathrm{m/s^2}$. Eight UAVs are located in the vicinity of the target. The initial state vectors of eight UAVs are $[x, y, z]^\intercal = [240, 240, 25]$, $[240, 260, 25]$, $[260, 240, 25]$, $[260, 260, 25]$, $[240, 250, 25]$, $[260, 250, 25]$, $[250, 240, 25]$ and $[250, 260, 25]$ with unit m. The rest of the parameters are the same as those set in Experiment 2. There are 65536 action combinations that consider the action space of all UAVs.

Fig. 17 is the CRLB of four control methods for 10 timesteps, which is averaged over 10 Monte Carlo experiments. Table IV is average and standard deviation of run-time to select one action combination. As shown in Fig. 17, The CRLB-based control has the lowest CRLB among the four control methods. The other three control methods achieve similar performance as the CRLB-based control, finding a sub-optimal action combination. Regarding run-time to select one action combination, the DRL-based control takes 0.02 seconds, which requires the least time. The CRLB-based control takes the most time to select an optimal solution by calculating the CRLB of all action combinations. The GA-based control and DPSO-based control select a sub-optimal action combination at every timestep and are advantageous in terms of run-time than CRLB-based control. Through the experiment in the larger-scale problem, it is confirmed that CRLB-based control can be replaced with alternative control methods: DRL-based control, DPSO-based control, and GA-based control, and that

DRL-based control is the most suitable method for real-time tracking while achieving performance close to the CRLB-based control.

VIII. Conclusions

It is expected that the multiple UAV control scheme for target tracking is essential for the SAR mission in the disaster environment. This paper has studied DRL-based multiple UAVs control to accurately track multiple FRs in the fields, decreasing CRLB of FRs state estimator. The state of each UAV is obtained by positions of other UAVs and targets. According to the trained Q-network, each of the UAVs selects its action control (i.e., flight direction). We exploited a reward function consisting of global reward and difference reward to quantify the effectiveness of the selected actions. Simulation results demonstrate that the proposed DRL-based multiple UAVs control is an algorithm that can replace CRLB-based control, which is advantageous for real-time tracking. For targets with various paths, DRL-based control enables multiple UAVs to track/localize target position accurately by improving the performance of the target state estimator. Besides, UAVs maintain an elevation angle between UAVs and targets to ensure sufficient the LoS probability from targets. DRL-based control achieves low CRLB comparable to that of the CRLB-based control and requires run-time at least 56x faster than CRLB-based control.

References

[1] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 36–42, May 2016.

[2] 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 Veh. Technol. Mag.*, vol. 12, no. 3, pp. 73–82, Sep. 2017.

[3] S. Zhang, H. Zhang, B. Di, and L. Song, "Cellular UAV-to-X communications: Design and optimization for multi-UAV networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 2, pp. 1346–1359, Feb. 2019.

[4] S. Zhang, H. Zhang, Q. He, K. Bian, and L. Song, "Joint trajectory and power optimization for UAV relay networks," *IEEE Commun. Lett.*, vol. 22, no. 1, pp. 161–164, Jan. 2018.

[5] N. Namvar, A. Homaifar, A. Karimoddini, and B. Maham, "Heterogeneous UAV cells: An effective resource allocation scheme for maximum coverage performance," *IEEE Access*, vol. 7, pp. 164 708–164 719, 2019.

[6] H. Huang and A. V. Savkin, "An algorithm of reactive collision free 3-D deployment of networked unmanned aerial vehicles for surveillance and monitoring," *IEEE Trans. Ind. Informat.*, vol. 16, no. 1, pp. 132–140, 2020.

[7] K. Peng, J. Du, F. Lu, Q. Sun, Y. Dong, P. Zhou, and M. Hu, "A hybrid genetic algorithm on routing and scheduling for vehicle-assisted multi-drone parcel delivery," *IEEE Access*, vol. 7, pp. 49 191–49 200, 2019.

[8] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, "Vehicle routing problems for drone delivery," *IEEE Trans. Syst., Man, Cybern. A, Syst.*, vol. 47, no. 1, pp. 70–85, 2017.

[9] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz, "Help from the sky: leveraging UAVs for disaster management," *IEEE Pervasive Comput.*, vol. 16, no. 1, pp. 24–32, Jan.-Mar. 2017.

[10] P. Rudol and P. Doherty, "Human body detection and geolocalization for UAV search and rescue missions using color and thermal imagery," in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, Mar. 2008, pp. 1–8.

[11] C. Yuan, Z. Liu, and Y. Zhang, "Fire detection using infrared images for UAV-based forest fire surveillance," in *Proc. IEEE Int. Conf. Unmanned Aircraft Syst. (ICUAS)*, Miami, FL, USA, Jun. 2017, pp. 567–572.

[12] W. Zhang, K. Song, X. Rong, and Y. Li, "Coarse-to-fine UAV target tracking with deep reinforcement learning," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 4, pp. 1522–1530, Oct. 2019.

[13] M. Wan, G. Gu, W. Qian, K. Ren, X. Maldague, and Q. Chen, "Unmanned aerial vehicle video-based target tracking algorithm using sparse representation," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9689–9706, 2019.

[14] C. G. Prevost, A. Desbiens, and E. Gagnon, "Extended kalman filter for state estimation and trajectory prediction of a moving object detected by an unmanned aerial vehicle," in *Proc. IEEE American Control Conf. (ACC)*, 2007, pp. 1805–1810.

[15] 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 *Proc. IEEE Int. Conf. Control Autom. (ICCA)*, 2010, pp. 273–278.

[16] Y. Sung and P. Tokekar, "GM-PHD filter for searching and tracking an unknown number of targets with a mobile sensor with limited FOV," *arXiv preprint arXiv:1812.09636*, 2018.

[17] T. Furukawa, F. Bourgault, B. Lavis, and H. F. Durrant-Whyte, "Recursive bayesian search-and-tracking using coordinated UAVs for lost targets," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Orlando, FL, May 2006, pp. 2521–2526.

[18] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 41–53, July 2005.

[19] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[20] Y. Zhang, Z. Zhuang, F. Gao, J. Wang, and Z. Han, "Multi-agent deep reinforcement learning for secure UAV communications," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2020, pp. 1–5.

[21] Q. Wang, W. Zhang, Y. Liu, and Y. Liu, "Multi-UAV dynamic wireless networking with deep reinforcement learning," *IEEE Commun. Lett.*, vol. 23, no. 12, pp. 2243–2246, Dec. 2019.

[22] C. H. Liu, Z. Chen, J. Tang, J. Xu, and C. Piao, "Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2059–2070, Sep. 2018.

[23] D. Chen, Q. Qi, Z. Zhuang, J. Wang, J. Liao, and Z. Han, "Mean field deep reinforcement learning for fair and efficient UAV control," *IEEE Internet Things J.*, 2020.

[24] S. Bhagat and S. PB, "UAV target tracking in urban environments using deep reinforcement learning," *arXiv preprint arXiv:2007.10934*, 2020.

[25] B. Li and Y. Wu, "Path planning for UAV ground target tracking via deep reinforcement learning," *IEEE Access*, vol. 8, pp. 29 064–29 074, 2020.

[26] C. Wang, J. Wang, Y. Shen, and X. Zhang, "Autonomous navigation of UAVs in large-scale complex environments: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2124–2136, Mar. 2019.

[27] C. Wang, J. Wang, J. Wang, and X. Zhang, "Deep-reinforcement-learning-based autonomous UAV navigation with sparse rewards," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6180–6190, 2020.

[28] X. Li, H. Yao, J. Wang, S. Wu, C. Jiang, and Y. Qian, "Rechargeable multi-uav aided seamless coverage for qos-guaranteed iot networks," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10 902–10 914, 2019.

[29] X. Li, H. Yao, J. Wang, X. Xu, C. Jiang, and L. Hanzo, "A near-optimal uav-aided radio coverage strategy for dense urban areas," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 9098–9109, 2019.

[30] H. Binol, E. Bulut, K. Akkaya, and I. Guvenc, "Time optimal multi-uav path planning for gathering its data from roadside units," in *Proc. IEEE Veh. Technol. Conf. (VTC-Fall)*, 2018, pp. 1–5.

[31] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[32] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[33] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot and N. Freitas, "Dueling network architectures for deep reinforcement learning," *arXiv preprint arXiv:1511.06581*, 2015.

[34] Z. R. Zaidi and B. L. Mark, "Mobility tracking based on autoregressive models," *IEEE Trans. Mobile Comput.*, vol. 10, no. 1, pp. 32–43, 2011.

[35] L. Mihaylova, D. Angelova, S. Honary, D. R. Bull, C. N. Canagarajah, and B. Ristic, "Mobility tracking in cellular networks using particle filtering," *IEEE Trans. Wireless Commun.*, vol. 6, no. 10, pp. 3589–3599, 2007.

[36] 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 *Proc. IEEE Int. Conf. Unmanned Aircraft Syst. (ICUAS)*, 2020, pp. 1475–1484.

[37] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Dec. 2014.

[38] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[39] I. Y. Kim and O. L. de Weck, "Adaptive weighted sum method for multiobjective optimization: a new method for pareto front generation," *Struct. Multidiscipl. Optim.*, vol. 31, no. 2, pp. 105–116, 2006.

[40] J. Wang, C. Jiang, H. Zhang, Y. Ren, K. C. Chen, and L. Hanzo, "Thirty years of machine learning: The road to pareto-optimal wireless networks," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1472–1514, 2020.

[41] A. K. Agogino and K. Tumer, "Analyzing and visualizing multiagent rewards in dynamic and stochastic domains," *Auton. Agents Multi-Agent Syst.*, vol. 17, no. 2, pp. 320–338, 2008.

[42] M. Colby and K. Turner, "Fitness function shaping in multiagent cooperative coevolutionary algorithms," *Auton. Agent Multi-Agent Syst.*, vol. 31, no. 2, pp. 179–206, 2017.

[43] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman filter: Particle filters for tracking applications*. Artech house, 2003.

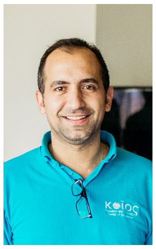
[44] D. E. Golberg, *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, 1989.

[45] W. Chen, J. Zhang, H. S. H. Chung, W. Zhong, W. Wu, and Y. Shi, "A novel set-based particle swarm optimization method for discrete optimization problems," *IEEE Trans. Evol. Comput.*, vol. 14, no. 2, pp. 278–300, 2010.

Jiseon Moon received her B.S. degree in Information Communication Engineering from Inha University of Incheon, South Korea, in 2019. She is currently pursuing the combined master's and Ph.D. degrees in the Department of Electronics and Computer Engineering from Hanyang University, Seoul, South Korea. Her research interests include wireless localization/positioning systems, multi-target tracking and location-aware communications

Savvas Papaioannou obtained his B.S. degree in Electronic and Computer Engineering from Technical University of Crete, Greece, his M.S. degree in Electrical Engineering from Yale University, USA and his Ph.D. degree in Computer Science from the University of Oxford, UK. He is currently a Research Associate at the KIOS Research and Innovation Center of Excellence at the University of Cyprus. His research interests include multi-agent and autonomous systems, state estimation and control, multi-target tracking, probabilistic inference, Bayesian reasoning and intelligent UAV systems and applications. He is a member of the IEEE and the ACM and also a reviewer for various journals and conferences within the IEEE and ACM associations.

Christos Laoudias is a Research Lecturer at KIOS Research and Innovation Center of Excellence (CoE), University of Cyprus leading various projects and activities related to localization, tracking, and navigation in wireless networks. Before that he was leading the geolocation technology group in Huawei Ireland Research Center. He holds a Diploma in Computer Engineering and Informatics (2003) and a M.Sc. in Integrated Hardware and Software Systems (2005) from the University of Patras, Greece, and a Ph.D. in Computer Engineering from the University of Cyprus (2014). During his doctoral studies and later as a postdoctoral fellow with KIOS CoE he coached the development of several award-winning indoor localization prototype systems, which have been released under open-source license. His research interests include positioning and tracking technologies, mobile and pervasive location-awareness, fault-tolerant location estimation, and location-based services.



Panayiotis Kolios received the BEng and PhD degrees in telecommunications Engineering from King's College London, in 2008 and 2011, respectively. Currently he is Research Assistant Professor at the KIOS Research and Innovation Center of Excellence, University of Cyprus. His interests focus on both basic and applied research on networked intelligent systems. Some examples of such systems include intelligent transportation systems, autonomous unmanned aerial systems, and the plethora of cyber-physical systems that arise within the Internet of Things. Particular emphasis is given to emergency response aspects in which faults and attacks could cause disruptions that need to be effectively handled. He is an active member of the IEEE, contributing to a number of technical and professional activities within the Association.

Sunwoo Kim (S'99-M'05-SM'17) received his B.S degree from Hanyang University, Seoul, Korea in 1999, and his Ph.D. degree, in 2005, from the Department of Electrical and Computer Engineering, University of California, Santa Barbara. Since 2005, he has been working in the Department of Electronic Engineering at Hanyang University, Seoul, Korea, where he is currently a professor. He is also the director of the 5G/Unmanned Vehicle Research Center, funded by the Ministry of Science and ICT of Korea. He was a visiting scholar to the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology from 2018 to 2019. He is an associate editor of IEEE Transactions on Vehicular Technology. He is a senior member of the IEEE. His research interests include wireless communication/positioning/localization, signal processing, vehicular networks, and location-aware communications.