UAV-Assisted Enhanced Coverage and Capacity in Dynamic MU-mMIMO IoT Systems: A Deep Reinforcement Learning Approach

MohammadMahdi Ghadaksaz, Mobeen Mahmood, Tho Le-Ngoc Department of Electrical and Computer Engineering McGill University, Montreal, QC, Canada Email: mohammad.ghadaksaz@mail.mcgill.ca, mobeen.mahmood@mail.mcgill.ca, tho.le-ngoc@mcgill.ca

Abstract—This study focuses on a multi-user massive multipleinput multiple-output (MU-mMIMO) system by incorporating an unmanned aerial vehicle (UAV) as a decode-and-forward (DF) relay between the base station (BS) and multiple Internet-of-Things (IoT) devices. Our primary objective is to maximize the overall achievable rate (AR) by introducing a novel framework that integrates joint hybrid beamforming (HBF) and UAV localization in dynamic MU-mMIMO IoT systems. Particularly, HBF stages for BS and UAV are designed by leveraging slow time-varying angular information, whereas a deep reinforcement learning (RL) algorithm, namely deep deterministic policy gradient (DDPG) with continuous action space, is developed to train the UAV for its deployment. By using a customized reward function, the RL agent learns an optimal UAV deployment policy capable of adapting to both static and dynamic environments. The illustrative results show that the proposed DDPG-based UAV deployment (DDPG-UD) can achieve approximately 99.5% of the sum-rate capacity achieved by particle swarm optimization (PSO)-based UAV deployment (PSO-UD), while requiring a significantly reduced runtime at approximately 68.50% of that needed by PSO-UD, offering an efficient solution in dynamic MU-mMIMO environments.

I. Introduction

In the next-generation of wireless communications, the expectation of connectivity and in the second connectivity and in the pectation of connectivity anywhere and anytime poses a formidable challenge, particularly in dynamic Internet-of-Things (IoT) environments where the users are continuously on the move, changing their locations frequently. These IoT environments, spanning from ever-changing urban areas to critical emergencies, necessitate a network infrastructure that is both adaptable and robust. In this case, several networking strategies have been explored, such as direct transmission and relay-based configurations. However, direct transmission over large distances can be impractical and result in excessive power consumption. It also frequently fails to fulfill the dynamic demands and changing conditions of these areas when employing the traditional static structure of base station (BS). Under these conditions, employing mobile relay nodes emerges as a more energy-efficient solution [1].

Recent developments in unmanned aerial vehicles (UAVs), commonly referred to as drones, have positioned them as an essential element of the future wireless communications networks. When used as relays, UAVs offer several advantages over traditional static relay systems. Specifically, mobile, ondemand relay systems are highly suitable for unforeseen or transient events, like emergencies or network offloading tasks, due to their ability to be quickly and economically deployed [2]. The mobility of UAVs enables them to operate at relatively high elevations, set up a line-of-sight connection with users on the ground, and prevent signal interference caused by obstacles, which makes UAVs practically appealing for dynamic communications systems [3]. Despite the significant propagation challenges faced by millimeter-wave (mmWave) signals, including free-space path loss, atmospheric and molecular absorption, and attenuation from rain, their substantial bandwidth presents a promising solution for meeting the high-throughput and low-latency requirements of diverse UAV application scenarios [4]. To address these challenges, massive multiple-input multiple-output (mMIMO) technology is employed, utilizing large antenna arrays to generate robust beam signals and thereby extending the transmission range. Compared to fullydigital beamforming (FDBF), the hybrid beamforming (HBF) architecture, which consists of a radio frequency (RF) stage and a baseband (BB) stage, can minimize power consumption by reducing the number of energy-consuming RF chains while achieving a performance close to FDBF [5]-[7].

The deployment of the UAV plays an important role in enhancing the performance of UAV-assisted wireless communications systems. Thus, recent research has shown an increased interest in optimizing UAV locations, with particular emphasis on HBF solutions to maximize the achievable rate (AR) or minimize transmit power [8]-[11]. In particular, [8] investigates the joint optimization of UAV deployment while considering HBF at BS and UAV for maximum AR. An amply-and-forward UAV relay with analog beamforming architecture is considered in [9] to maximize the capacity in a dual-hop mMIMO IoT system. Similarly, [12] considers the optimization problem for UAV location, user clustering, and HBF design to maximize AR under a minimum rate constraint for each user. The authors in [13] study the joint optimization of UAVs flying altitude, position, transmit power, antenna beamwidth, and users' allocated bandwidth. Most of the existing research works (e.g., [8]-[13]) tackle the issue of UAV deployment in a static environment where users are situated at fixed locations. However, these works tend to overlook the dynamic nature of real-world environments, where ground IoT users/devices exhibit mobility, leading to rapidly changing conditions.

As a cornerstone of artificial intelligence (AI), reinforcement learning (RL) has been extensively researched in wireless communications and UAV applications [14]-[16]. RL is a decision-making approach that emphasizes learning through interaction with an environment. Inspired by behavioral psychology, RL is akin to the learning process in humans and animals, where actions are taken based on past experiences and their outcomes. Within the spectrum of RL methods, the deep deterministic policy gradient (DDPG) has emerged as a notable technique [17]. DDPG is particularly adept at handling continuous action spaces, which are common in real-world scenarios. This capability makes DDPG highly relevant for complex, dynamic environments where actions need to be precise and varied, as is often the case with UAV operations. Different RL-based solutions have been studied for UAV positioning (e.g., [18]-[20]). However, the design of HBF jointly with UAV deployment using RL-based solutions is an unaddressed research problem, presenting a significant opportunity to advance the field of UAV-assisted mMIMO IoT communications networks in dynamic environments.

To address this issue, we propose a joint HBF and UAV deployment framework using the DDPG-based algorithmic solution to maximize AR in dynamic MU-mMIMO IoT systems. In particular, the RF beamforming stages for BS and UAV are designed based on the slow time-varying angle-of-departure (AoD)/angle-of-arrival (AoA) information, and BB stages are formulated using the reduced-dimensional effective channel matrices. Then, a novel DDPG-based algorithmic solution is proposed for UAV deployment with a primary objective to not only maximize the overall AR in MU-mMIMO IoT systems but also to significantly reduce computational complexity, particularly the runtime, compared to nature-inspired (NI) optimization methods. It is worthwhile to mention that the proposed DDPG algorithm exhibits a notable advantage in dynamic scenarios. The knowledge gained from training on the initial user's position is effectively transferred to subsequent positions, resulting in reduced learning time—akin to transfer learning. The illustrative results depict the efficacy of the proposed DDPG-based deployment scheme by reducing the runtime up to 31.5% as compared to NI-based solutions.

The rest of this paper is organized in the following manner. Section II defines the system and channel model for UAV-relay MU-mMIMO systems. In Section III, we introduce the joint HBF and DDPG-based UAV deployment framework. Section IV presents the illustrative results. Finally, Section V concludes the paper.

II. SYSTEM & CHANNEL MODEL

A. System Model

The current research explores a complex situation in which various users are linked to a gateway via both wired and wireless connections. This configuration is located in a remote region, which is hard to reach directly by BS because of several hindrances like buildings, mountains, etc. Afterward, a UAV is employed as a dual-hop decode-and-forward (DF) relay to connect with the users as illustrated in Fig. 1. Let (x_b, y_b, z_b) , (x_u, y_u, z_u) , and (x_k, y_k, z_k) denote the location of the BS, UAV relay, and k^{th} IoT user, respectively. We establish the 3D distances for a UAV-assisted mmWave MU-mMIMO IoT system as follows:

$$\tau_{1} = \sqrt{(x_{u} - x_{b})^{2} + (y_{u} - y_{b})^{2} + (z_{u} - z_{b})^{2}}$$

$$\tau_{2,k} = \sqrt{(x_{u} - x_{k})^{2} + (y_{u} - y_{k})^{2} + (z_{u} - z_{k})^{2}}$$

$$\tau_{k} = \sqrt{(x_{b} - x_{k})^{2} + (y_{b} - y_{k})^{2} + (z_{b} - z_{k})^{2}}$$
(1)

where τ_1 , $\tau_{2,k}$, and τ_k are the 3D distance between UAV & BS, between the UAV and k^{th} IoT user, and between BS and k^{th} IoT user, respectively.

In this system model, we consider BS equipped with N_T antennas, UAV relay with N_r antennas for receiving and N_t antennas for sending information to K single-antenna IoT users scattered in G groups, where g^{th} group has K_g IoT users such that $K = \sum_{g=1}^G K_g$. Both BS and UAV relay utilize HBF architecture. In this setup, the BS includes

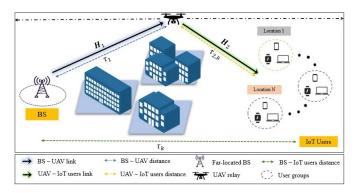


Fig. 1. UAV-assisted mmWave MU-mMIMO dynamic environment

an RF beamforming stage $\mathbf{F}_b \in \mathbb{C}^{N_T \times N_{RF_b}}$ and BB stage $\mathbf{B}_b \in \mathbb{C}^{N_{RF_b} \times K}$, where N_{RF_b} is the number of RF chains such that $N_s \leq N_{RF_b} \leq N_T$ to guarantee multi-stream transmission. Considering half-duplex (HD) DF relaying, BS sends K data streams $\mathbf{d} = [d_1, d_2, \cdots, d_K]^T$ through channel $\mathbf{H}_1 \in \mathbb{C}^{N_r \times N_T}$ in the first time slot. Using N_r antennas, UAV receives signals with RF stage $\mathbf{F}_{u,r} \in \mathbb{C}^{N_{RF_u} \times N_r}$ and BB stage $\mathbf{B}_{u,r} \in \mathbb{C}^{K \times N_{RF_u}}$. We assume UAV relay transmits the data in the second time slot using RF beamformer $\mathbf{F}_{u,t} = [\mathbf{f}_{u,t,1} \cdots, \mathbf{f}_{u,t,N_{RF_u}}] \in \mathbb{C}^{N_t \times N_{RF_u}}$ and BB stage $\mathbf{B}_{u,t} = [\mathbf{b}_{u,t,1}, \cdots, \mathbf{b}_{u,t,K}] \in \mathbb{C}^{N_{RF_u} \times K}$ via channel $\mathbf{H}_2 \in \mathbb{C}^{K \times N_t}$. The HBF design significantly cuts down the number of RF chains, for example, reducing them from N_T RF chains to N_{RF_u} for BS, and from $N_t(N_r)$ RF chains to N_{RF_u} for UAV, while satisfying the following conditions: 1) $K \leq N_{RF_b} \ll N_T$; and 2) $K \leq N_{RF_u} \ll N_r(N_t)$. In this case, considering the environment noise follows the distribution $\mathcal{CN}(\mathbf{0}, \sigma_n^2)$, the AR for the BS-UAV link can be expressed as follows [21]:

$$\mathbf{R}_{1}(\mathbf{F}_{b}, \mathbf{B}_{b}, \mathbf{F}_{u,r}, \mathbf{B}_{u,r}) = \log_{2} |\mathbf{I}_{K} + \mathbf{Q}_{1}^{-1}\mathbf{B}_{u,r}\mathcal{H}_{1}\mathbf{B}_{b}\mathbf{B}_{b}^{H}\mathcal{H}_{1}^{H}\mathbf{B}_{u,r}^{H}|,$$
 (2) where $\mathbf{Q}_{1}^{-1} = (\sigma_{n}^{2}\mathbf{B}_{u,r}\mathbf{F}_{u,r})^{-1}\mathbf{F}_{u,r}^{H}\mathbf{B}_{u,r}^{H}, \mathcal{H}_{1} = \mathbf{F}_{u,r}\mathbf{H}_{1}\mathbf{F}_{b},$ and $\mathbb{E}\{\mathbf{dd}^{H}\} = \mathbf{I}_{K} \in \mathbb{C}^{K \times K}$. In the same manner, the AR between UAV and IoT users is calculated based on the instantaneous signal-to-interference-plus-noise ratio (SINR) as demonstrated by the following expression [21]:

demonstrated by the following expression [21]:
$$SINR_{g_k} = \frac{|\mathbf{h}_{2,k}^H \mathbf{F}_{u,t} \mathbf{b}_{u,t,g_k}|^2}{\sum_{\hat{k} \neq k}^{Kg} |\mathbf{h}_{2,k}^H \mathbf{F}_{u,t} \mathbf{b}_{u,t,g_{\hat{k}}}|^2 + \sum_{q \neq g}^{G} \sum_{\hat{k} \neq k}^{Kg} |\mathbf{h}_{2,k}^H \mathbf{F}_{u,t} \mathbf{b}_{u,t,q_{\hat{k}}}|^2 + \sigma_n^2}.$$
(3)

where $g_k = k + \sum_{g'=1}^{g-1} K_{g'}$ represents the IoT user index and $\mathbf{h}_{2,g_k} \in \mathbb{C}^{N_t}$ is the channel vector between UAV and respective IoT user. Utilizing the instantaneous SINR, the ergodic AR of the second link R_2 , in UAV-assisted mmWave MU-mMIMO systems, can be written as:

R₂(
$$\mathbf{F}_{u,t}, \mathbf{B}_{u,t}, x_u, y_u$$
) = $\mathbb{E}\left\{\sum_{g=1}^{G} \sum_{k=1}^{K_g} \mathbb{E}\left[\log_2(1 + \text{SINR}_{g_k})\right]\right\}$.

(4)

We consider the mmWave channel for both links, then using the Saleh-Valenzuela channel model, the channel between BS and UAV can be written as follows [22]:

$$\mathbf{H}_{1} = \sum_{c=1}^{C} \sum_{l=1}^{L} z_{1_{cl}} \tau_{l_{cl}}^{-\eta} \mathbf{a}_{1}^{(r)}(\theta_{cl}^{(r)}, \phi_{cl}^{(r)}) \mathbf{a}_{1}^{(t)T}(\theta_{cl}^{(t)}, \phi_{cl}^{(t)})$$

$$= \mathbf{A}_{1}^{(r)} \mathbf{Z}_{1} \mathbf{A}_{1}^{(t)},$$
(5)

where C is the total number of groups, L is the total number of paths from BS to UAV, $z_{1_{cl}} \sim \mathcal{CN}(\mathbf{0}, \frac{1}{L})$ is the complex gain

of l^{th} path in the c^{th} cluster, and η is the path loss exponent. In addition, $\mathbf{a}_1^{(k)}(.,.)$ represents the respective transmit or receive array steering vector for a uniform rectangular array (URA), which is defined as [7]:

$$\mathbf{a}_{1}^{(k)}(\theta,\phi) = [1, e^{-j2\pi d\sin(\theta)\cos(\phi)}, \cdots, e^{-j2\pi d(N_{x}-1)\sin(\theta)\cos(\phi)}] \otimes (6)$$

$$[1, e^{-j2\pi d\sin(\theta)\sin(\phi)}, \cdots, e^{-j2\pi d(N_{y}-1)\sin(\theta)\sin(\phi)}],$$

where $k=\{r,t\}$, N_x and N_y are the horizontal and vertical size of respective antenna array at BS and UAV, d is the interelement spacing, $\mathbf{Z}_1=\mathrm{diag}(z_{1,1}\tau_{1,1}^{-\eta},...,z_{1,L}\tau_{1,L}^{-\eta})\in\mathbb{C}^{L\times L}$ represents the diagonal gain matrix, $\mathbf{A}_1^{(r)}\in\mathbb{C}^{N_r\times L}$ and $\mathbf{A}_1^{(t)}\in\mathbb{C}^{L\times N_t}$ are the receive and transmit phase response matrices, respectively. Also, the angles $\theta_{cl}^{(t)}\in[\theta_c^{(t)}-\delta_c^{\theta(t)},\theta_c^{(t)}+\delta_c^{\theta(t)}]$ and $\phi_{cl}^{(t)}\in[\phi_c^{(t)}-\delta_c^{\phi(t)},\phi_c^{(t)}+\delta_c^{\phi(t)}]$ denote the elevation AoD (EAoD) and azimuth AoD (AAoD) for l^{th} path in channel \mathbf{H}_1 , respectively. $\theta_c^{(t)}$ represent the mean EAoD and $\delta_c^{\theta(t)}$ is the EAoD spread, while $\phi_c^{(t)}$ is mean AAoD with spread $\delta_c^{\phi(t)}$. In a similar fashion, the angles $\theta_{cl}^{(r)}$ within the range $[\theta_c^{(r)}-\delta_c^{\theta(r)},\theta_c^{(r)}+\delta_c^{\theta(r)}]$ and $\phi_{cl}^{(r)}$ within $[\phi_c^{(r)}-\delta_c^{\phi(r)},\phi_c^{(r)}+\delta_c^{\phi(r)}]$ correspond to the elevation AoA (EAoA) and azimuth AoA (AAoA), respectively. Here, $\theta_c^{(r)}$ and $\phi_c^{(r)}$ represent the mean EAoA and AAoA, with $\delta_c^{\theta(r)}$ and $\delta_c^{\phi(r)}$ denoting the angular spreads of the elevation and azimuth angles, respectively. The channel vector between UAV and k^{th} IoT user is written as:

$$\mathbf{h}_{2,k}^{T} = \sum_{q=1}^{Q} z_{2,k_q} \tau_{2,k_q}^{-\eta} \mathbf{a}(\theta_{k_q}, \phi_{k_q}) = \mathbf{z}_{2,k}^{T} \mathbf{A}_{2,k} \in \mathbb{C}^{N_t}, \quad (7)$$

where Q is the total number of downlink paths from UAV to users, $z_{2,k_q} \sim \mathcal{CN}(0,\frac{1}{Q})$ is the complex path gain of q^{th} path in the second link, and $\mathbf{a}(.,.) \in \mathbb{C}^{N_t}$ is the UAV downlink array phase response vector. As given in (7), the obtained downlink channel is comprised of two distinct parts: 1) a fast time-varying path gain vector $z_{2,k} = [z_{2,k_1}\tau_{2,k_1}^{-\eta},...,z_{2,k_Q}\tau_{2,k_Q}^{-\eta}]^T \in \mathbb{C}^Q$; and 2) a slow time-varying downlink array phase response matrix $\mathbf{A}_{2,k} \in \mathbb{C}^{Q \times N_t}$ where each row is constituted by $\mathbf{a}(\theta_{kl},\phi_{kl})$. Afterward, the channel matrix for the second link can be expressed as follows:

$$\begin{split} \dot{\mathbf{H}}_2 &= [\mathbf{h}_{2,1}, \cdots, \mathbf{h}_{2,K}]^T = \mathbf{Z}_2 \mathbf{A}_2 \in \mathbb{C}^{K \times N_t}, \\ \text{where } \mathbf{Z}_2 &= [\mathbf{z}_{2,1}, \cdots, \mathbf{z}_{2,K}]^T \in \mathbb{C}^{K \times Q} \text{ is the complete path gain matrix for all downlink IoT users.} \end{split} \tag{8}$$

III. JOINT HBF & DDPG-BASED UAV DEPLOYMENT

In this section, our objective is to jointly optimize the UAV location and HBF for BS and UAV to reduce the channel state information (CSI) overhead size while maximizing the total AR of UAV-assisted MU-mMIMO IoT systems. First, we design the stages $\mathbf{F}_b, \mathbf{F}_{u,r}, \mathbf{F}_{u,t}$ based on the slow timevarying AoD and AoA. Then, the BB stages $\mathbf{B}_b, \mathbf{B}_{u,r}, \mathbf{B}_{u,t}$ are developed by using singular value decomposition (SVD).

A. HBF Design

The RF and BB stages for BS and UAV are designed for the following considerations: 1) maximize the beamforming gain at the desired directions based on the slow time-varying AoD and AoA; 2) reduce the power-hungry RF chains; 3) reduce the CSI overhead; and 4) mitigate multi-user interference (MU-I)¹.

B. DDPG: Preliminaries

DDPG is a sophisticated RL algorithm that combines elements of deep Q-networks (DQN) and policy gradient techniques. Unlike DQN, which is designed for discrete action spaces, DDPG is tailored for continuous action spaces, common in real-world situations. This makes DDPG ideal for complex tasks that demand a spectrum of continuous action values, increasing its effectiveness in diverse and changing environments.

DDPG utilizes actor-critic approach [17], where the actor, denoted as $\mu(\mathbf{s}|\theta^{\mu})$, outputs a deterministic action a given a state s, where θ^{μ} represents the weights of the actor network. In the same manner, critic expressed as $Q(\mathbf{s}, \mathbf{a} | \theta^Q)$, predicts the expected return (value) of taking an action a in a state s, and it is parameterized by weights θ^Q . DDPG uses target networks for both actor and critic, represented as $\mu'(\mathbf{s}|\theta^{\mu'})$ and $Q'(\mathbf{s}, \mathbf{a}|\theta^{Q'})$ respectively. Here, $\theta^{\mu'}$ and $\theta^{Q'}$ denote the weights of the target actor and target critic networks. These target networks are essentially slow-paced versions of the primary actor and critic networks, ensuring a more stable and consistent learning process. Moreover, to break the correlation between consecutive experiences (s_t , a_t , r_t , s_{t+1}), we use a replay buffer D, where s_t , a_t , r_t are the state, action, and reward at timestep t, respectively, and s_{t+1} is the next state. These transitions are stored in the replay buffer, and then at each timestep, the actor and critic are updated by sampling a minibatch of transitions $(\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}_{i+1})$ uniformly from the buffer. During the training, the weights of the critic are updated by minimizing the mean square error (MSE) loss function as:

$$\mathcal{L} = \frac{1}{N} \sum_{i} (y_i - Q(\mathbf{s}_i, \mathbf{a}_i | \theta^Q))^2, \tag{9}$$

where N is the batchsize and y_i is defined as:

$$y_i = r_i + \gamma Q'(\mathbf{s}_{i+1}, \mu'(\mathbf{s}_{i+1}|\theta^{\mu'})|\theta^{Q'}).$$
 (10)

Here, $\gamma \in [0,1]$ represents the discounting factor. Then, the actor network is updated using the policy gradient method as:

$$\nabla_{\theta^{\mu}} J \approx \frac{1}{N} \sum_{i} \nabla_{a} Q(\mathbf{s}, \mathbf{a} | \theta^{Q})|_{\mathbf{s} = \mathbf{s}_{i}, a = \mu_{\mathbf{s}_{i}}} \nabla_{\theta^{\mu}} \mu(\mathbf{s} | \theta^{\mu})|_{\mathbf{s}_{i}}.$$
(11)

Afterward, the target actor and target critic networks are updated using the soft update approach:

$$\theta^{Q'} \leftarrow \tau \theta^Q + (1 - \tau)\theta^{Q'} \tag{12}$$

$$\theta^{\mu'} \leftarrow \tau \theta^{\mu} + (1 - \tau)\theta^{\mu'} \tag{13}$$

where τ is a small coefficient denoting how fast the target actor and critic are updated. As in every RL algorithm, we need exploration to achieve the best policy. In this regard, we add a zero-mean Gaussian noise $\mathcal N$ to the actor policy as follows:

$$\mathbf{a}_t = \mu(\mathbf{s}_t | \theta_t^{\mu}) + \mathcal{N},\tag{14}$$

where \mathcal{N} follows the distribution $\mathcal{CN}(0, \sigma^2)$.

C. DDPG-Based UAV Deployment

To apply the DDPG algorithm to our problem, we need to define appropriate states, actions, and rewards for the agent. Moreover, the design of the actor and critic network can significantly influence the performance of the algorithm. Thus, first, we introduce the suitable state, action, and reward for UAV deployment, and then, we discuss the configuration of the actor and critic network.

¹The details for HBF design for BS and UAV can be found in [21]

Algorithm 1: DDPG-Based UAV Deployment

```
1 Randomly initialize actor \mu(\mathbf{s}|\theta^{\mu}) and critic
     Q(\mathbf{s}, \mathbf{a}|\theta^Q) networks with weights \theta^{\mu} and \theta^Q.
2 Initialize target network Q' and \mu' with weights
     \theta^{Q'} \leftarrow \theta^Q, \, \theta^{\mu'} \leftarrow \theta^\mu.
3 Initialize replay buffer D.
4 for episode=1:M do
        Receive initial observation state s_1.
5
        for t=1:T do
 6
             Select action \mathbf{a}_t using (14).
 8
             Execute action \mathbf{a}_t and observe reward r_t and
               new state s_{t+1}.
             Store transition (\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1}) in D.
             Sample a random minibatch of N transitions
10
               (\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}_{i+1}) from D.
             Formulate y_i using (10).
11
12
             Update the critic by minimizing the loss
               function \mathcal{L} in (9).
             Update the actor using the policy gradient in
13
             Update the target networks via (12), (13).
14
15
        end
16 end
```

1) States: At each time step t, the UAV agent will observe state s_t as follows:

$$\mathbf{s}_t = [\hat{x}_{u,t}, \hat{y}_{u,t}]^T \in \mathbb{R}^2, \tag{15}$$

 $\mathbf{s}_t = [\hat{x}_{u,t}, \hat{y}_{u,t}]^T \in \mathbb{R}^2, \tag{15}$ where $\hat{x}_{u,t} = \frac{x_{u,t}}{x_{\max}}$ and $\hat{y}_{u,t} = \frac{y_{u,t}}{y_{\max}}$ denote the 2D normalized location of the UAV at time step t. Here, we assume the UAV is deployed at a fixed height $z_{u,t}^2$.

2) Actions: We consider action \mathbf{a}_t at time step t for the UAV agent as follows:

$$\mathbf{a}_{t} = [a_{t,x}, a_{t,y}]^{T} \in \mathbb{R}^{2}, \quad \left\{ \begin{array}{l} a_{t,x} \in [-a_{x,\max}, a_{x,\max}] \\ a_{t,y} \in [-a_{y,\max}, a_{y,\max}] \end{array} \right.$$
(16)
Here, $a_{x,\max}$ $(a_{y,\max})$ represents the maximum movement step

for the UAV on the x-axis (y-axis). When $a_{t,x}$ ($a_{t,y}$) is positive, the movement is to the East (North), and when $a_{t,x}$ ($a_{t,y}$) is negative, the movement is to the West (South).

3) Reward: The reward function effectively communicates the objectives of the task to the agent. Designing the reward function correctly is pivotal since it not only shapes the learning trajectory but also influences the convergence speed and the overall effectiveness of the policy learned. Considering this, we define the reward function r_t at the time step t as:

$$r_{t} = \begin{cases} R_{2} & \text{for } R_{2} \ge \eta_{0} \\ -1 & \text{for } R_{2} < \eta_{0} \\ -5 & \text{for } x_{u,t} > x_{\text{max}} \text{ or } y_{u,t} > y_{\text{max}} \\ -5 & \text{for } x_{u,t} < x_{\text{min}} \text{ or } y_{u,t} < y_{\text{min}} \end{cases}$$
(17)

where R₂ denotes the achievable rate for the second link as given in (4), η_0 is an adjustable threshold, x_{max} (x_{min}) is the maximum (minimum) allowable position on the x-axis, and y_{max} (y_{min}) is the maximum (minimum) permitted position on the y-axis.

TABLE I. SIMULATION PARAMETERS

| Number of antennas | | $(N_T, N_t, N_r) = 144$ | |
|--------------------------------|----------------------------------|------------------------------------|--|
| BS height | UAV height | 10 m | 20 m |
| UAV x-axis range | UAV y-axis range | $[x_{\min}, x_{\max}] = [0,100]m$ | $[y_{\min}, y_{\max}] = [0,100]\text{m}$ |
| UAV x-axis movement | UAV y-axis movement | $[a_{x,min}, a_{x,max}] = [-1,1]m$ | $[a_{y,min}, a_{y,max}] = [-1,1]m$ |
| User groups | # of users per group | G = 1 | $K_g = \frac{K}{G}$ |
| # of paths | Path loss exponent | L = 10 | 3.6 |
| Noise PSD | Reference path loss α | -174 dBm/Hz | 61.34 dB |
| Frequency | Channel bandwidth | 28 GHz | 100 MHz |
| Mean AAoD/AAoA (1st link) | Mean AAoD (2 nd link) | 120° | $\phi_g = 21^{\circ} + 120^{\circ}(g - 1)$ |
| Mean EAoD/EAoA (1st link) | Mean EAoD (2 nd link) | 60° | $\theta_g = 60^{\circ}$ |
| Azimuth/Elevation angle spread | # of network realization | ±10° | 2000 |

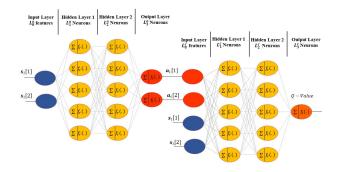


Fig. 2. (Target) actor and (target) critic DNN architecture

4) Actor and Critic Networks: We employ a fully connected deep neural network (DNN) architecture with two hidden layers as depicted in Fig. 2, for both (target) actor and (target) critic networks. Here, the actor network predicts suitable actions as described in (16) based on the input states given in (15), whereas, the critic network determines the Qvalue of the input action-state pair. We consider L_i^a neurons in each i^{th} hidden layer for the actor network with $i = \{1, 2\}$. Similarly, we will have L_j^c neurons in each j^{th} hidden layer for the critic network with $j = \{1,2\}$.

To perform non-linear operations, we utilize the rectified linear unit (ReLU) as the activation function in the hidden layers for both actor and critic networks (i.e., $f_r(z) = \max(0, z)$). To ensure that the predicted actions by the actor network are between $[-a_{x,\max}, a_{x,\max}]$ ($[-a_{y,\max}, a_{y,\max}]$), we must use a function that can output both negative and positive values. Thus, we apply tanh activation function in the output layer of the actor network (i.e., $f_t(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$). However, since critic network estimates the Q-value function, the range of outputs is quite large or unbounded. Therefore, we use the linear activation function for the output layer of the critic network (i.e., $f_l(z) = z$). The summary of the DDPG algorithm is outlined in Algorithm 1.

IV. ILLUSTRATIVE RESULTS

In this section, we present illustrative results on the performance of the DDPG-based UAV deployment (DDPG-UD) algorithm for different scenarios. For benchmark comparison, we compare the proposed DDPG-UD with the following solutions: 1) particle swarm optimization (PSO)-based UAV deployment (PSO-UD); and 2) deep learning (DL)-based UAV deployment (DL-UD), (i.e., supervised learning (SL) approach [21]). Table I outlines the simulation setup based on the 3D micro-cell scenario [7], whereas the hyper-parameters of the DDPG algorithm are given in Table II. In the following, we compare the performance for both static (fixed user locations) and dynamic (users changing locations) environments in MUmMIMO IoT systems.

²For simplicity, we consider a scenario of fixed UAV height. However, the proposed DDPG-based solution can be applied for 3D UAV deployment, which is left as our future work.

TABLE II. NETWORKS' PARAMETERS

| (Target) Actor Network Architecture | | | | | |
|--------------------------------------|--------------|------------------------------------|--------------|--|--|
| Input Shape | $L_0^a = 2$ | 1^{st} hidden layer $L_1^a = 2$ | | | |
| 2 nd hidden layer | $L_2^a = 20$ | Output layer | $L_3^a = 2$ | | |
| (Target) Critic Network Architecture | | | | | |
| Input Shape | $L_0^c = 4$ | 1 st hidden layer | $L_1^c = 20$ | | |
| 2 nd hidden layer | $L_2^c = 20$ | Output layer | $L_3^c = 1$ | | |
| Network Parameters | | | | | |
| Reply buffer size | 60000 | Critic learning rate 0.002 | | | |
| Actor learning rate | 0.001 | Target networks learning rate 0.01 | | | |

A. Static Environment (Fixed Users Location)

In this section, we consider IoT users to have a fixed location and discuss two scenarios: 1) narrow-range user distribution; and 2) wide-range user distribution. For narrowrange user distribution, we consider that the BS is located at $(x_b, y_b, z_b) = (0, 0, 10)$, the UAV initial position is $(x_u, y_u, z_u) = (50, 50, 20)$, and K = 4 IoT users are distributed randomly at a far distance from the BS (i.e., $(x_k, y_k) \in$ [90, 100]). UAV starts its initial location at (50, 50, 20) at the beginning of each episode, and then it explores the environment during the time steps. For wide-range user distribution, we consider the same location and initial position for the UAV, while assuming that K = 4 IoT users are randomly scattered with $(x_k, y_k) \in [50, 100]$. Fig. 3 shows the achieved rates for DDPG-UD, PSO-UD, DL-UD, and fixed deployment (FD) (i.e., no optimization). Numerical results reveal that the proposed DDPG-UD can achieve 98.63% of PSO-UD performance for narrow-range user distribution while having a 16 times better performance than FD. Furthermore, DDPG-UD can enhance the performance of PSO-UD by 2.23% for wide-range user distribution while having 3.14 times better AR than the FD. Fig. 3 also demonstrates that although DL-UD can have an acceptable performance when we have narrowrange user distribution while achieving 89.62% AR of PSO-UD, it fails to find the optimal location for wide-range user distribution having only 36.94% AR of the DDPG-UD. This means that for more complex user distributions, DL-UD is not a promising solution. For the next step, we consider that IoT users may change their location during the observation.

B. Dynamic Environment (Changing Users Locations)

In this section, we consider a more practical scenario where IoT users can change their location during the training. This scenario represents dynamic, real-world environments. We assume that the BS is set at $(x_b,y_b,z_b)=[0,0,10]$. Then we set the IoT users' locations randomly at six different distributions $l\in\{l_1,l_2,...,l_6\}$ as follows:

$$(x_{k,l}, y_{k,l}) = \begin{cases} x_{k,l} \in [60, 70], & y_{k,l} \in [60, 70] & \text{for } l_1 \\ x_{k,l} \in [60, 70], & y_{k,l} \in [70, 80] & \text{for } l_2 \\ x_{k,l} \in [70, 80], & y_{k,l} \in [80, 90] & \text{for } l_3 \\ x_{k,l} \in [80, 90], & y_{k,l} \in [80, 90] & \text{for } l_4 \\ x_{k,l} \in [80, 90], & y_{k,l} \in [70, 80] & \text{for } l_5 \\ x_{k,l} \in [80, 90], & y_{k,l} \in [60, 70] & \text{for } l_6 \end{cases}$$

$$(18)$$

where $x_{k,l}$ and $y_{k,l}$ denote the k^{th} IoT user x-axis and y-axis range during l^{th} distribution, respectively. Furthermore, UAV starts its initial location at $(x_u, y_u, z_u) = (50, 50, 20)$ and finds the optimal deployment $\mathbf{x}_o^{(1)} = \{x_o^{(1)}, y_o^{(1)}\}$ for l_1 , which is now used as its initial location for the next user distribution (i.e., l_2). Fig. 4 shows the accumulated reward and average accumulated reward for the DDPG agent during six different user locations. This figure shows that after the first

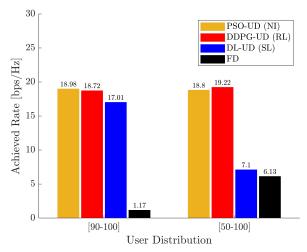


Fig. 3. Achieved Rates versus user distribution

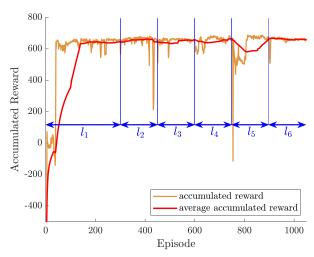


Fig. 4. Accumulated reward vs episode for DDPG

user location, the DDPG agent maintains the reward and finds the optimal location in less number of episodes. The reason is that the pre-trained networks from previous user locations contain information about the environment, thus helping the DDPG agent find the optimal UAV location for changing user locations in a shorter time. Fig. 5 shows the achieved rate for DDPG-UD, PSO-UD, DL-UD, and FD, which shows that DDPG-UD can achieve a performance close to PSO-UD. For instance, DDPG-UD accomplishes 99.62% of PSO-UD AR for l_3 , and it provides 18.54 bps/Hz AR at l_6 and achieves 99.42% of PSO-UD performance. Additionally, the capacity is improved by 36.99% and 16.73% for FD for l_3 and l_6 , respectively. Fig. 5 also supports the previous deduction that DL-UD is not a promising solution for complex scenarios while having only 63.84% and 66.65% of PSO-UD performance for l_3 and l_6 , respectively.

Fig. 6 displays the time comparison between DDPG-UD and PSO-UD. Here, we provide the runtime for a single user location (i.e., $l \in l_1$), three different user locations ($l \in \{l_1, l_2, l_3\}$), and six distinct user locations ($l \in \{l_1, \cdots, l_6\}$). This figure shows that although for a single user location, DDPG-UD takes 18.04% more than PSO-UD to find the optimal location, for multiple user locations, it takes 76.45% and 68.50% of PSO-UD runtime. This means that as the

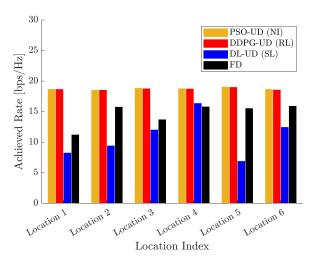


Fig. 5. Achieved rates in different locations

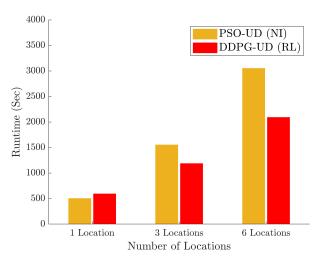


Fig. 6. Runtime for different numbers of locations

number of user locations is increased, DDPG-UD will take less time to find the optimal location when compared to PSO-UD, which makes DDPG-UD an efficient solution for dynamic and fast-changing environments in MU-mMIMO IoT systems.

V. CONCLUSION

In this work, a novel DDPG-based UAV deployment (DDPG-UD) and hybrid beamforming (HBF) technique has been proposed for AR maximization in dynamic MU-mMIMO IoT systems. First, we introduce HBF for both base station (BS) and UAV. Afterward, we apply the deep deterministic policy gradient (DDPG) as a reinforcement learning approach to UAV deployment in dynamic environments. Illustrative results show that the proposed DDPG-UD closely approaches the optimal rate achieved by particle swarm optimization (PSO)-based UAV deployment (PSO-UD). On the other hand, DDPG-UD greatly reduces the runtime by 31.5%, which makes DDPG-UD a more appropriate solution for real-world dynamic applications in UAV-assisted mMIMO IoT systems.

REFERENCES

 Y. Zeng et al., "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 54, no. 5, pp.

- 36-42, 2016.
- [2] M. Mozaffari et al., "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2334–2360, 2019.
- [3] Z. Xiao, P. Xia, and X.-G. Xia, "Enabling UAV cellular with millimeterwave communication: Potentials and approaches," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 66–73, 2016.
- [4] W. Roh et al., "Millimeter-wave beamforming as an enabling technology for 5G cellular communications: theoretical feasibility and prototype results," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 106–113, 2014.
- [5] M. Mahmood et al., "2D antenna array structures for hybrid massive MIMO precoding," in Proc. IEEE Global Commun. Conf. (GLOBE-COM), 2020, pp. 1-6.
- [6] M. Mahmood, A. Koc, and T. Le-Ngoc, "3-D antenna array structures for millimeter wave multi-user massive MIMO hybrid precoder design: A performance comparison," *IEEE Commun. Lett.*, vol. 26, no. 6, pp. 1393–1397, 2022.
- [7] M. Mahmood et al., "Energy-efficient MU-massive-MIMO hybrid precoder design: Low-resolution phase shifters and digital-to-analog converters for 2D antenna array structures," *IEEE Open J. Commun. Soc.*, vol. 2, no. 5, pp. 1842-1861, 2021.
- [8] M. Mahmood et al., "PSO-Based Joint UAV Positioning and Hybrid Precoding in UAV-Assisted Massive MIMO Systems" in Proc. IEEE 96th Veh. Technol. Conf. (VTC-Fall), 2022, pp. 1-6.
- [9] M. Mahmood et al., "Spherical Array-Based Joint Beamforming and UAV Positioning in Massive MIMO Systems" in Proc. IEEE 97th Veh. Technol. Conf. (VTC-Spring), 2023, pp. 1-5.
- [10] X. Xi et al., "Joint user association and UAV location optimization for UAV-aided communications," *IEEE Wireless Commun. Lett.*, vol. 8, no. 6, pp. 1688–1691, 2019.
- [11] M. Alzenad *et al.*, "3-D placement of an unmanned aerial vehicle base station (UAV-BS) for energy-efficient maximal coverage," *IEEE Wireless Commun. Lett.*, vol. 6, no. 4, pp. 434–437, 2017.
- [12] L. Zhu et al., "Multi-UAV aided millimeter-wave networks: Positioning, clustering, and beamforming," *IEEE Trans. Wireless Commun.*, vol. 21, no. 7, pp. 4637–4653, 2022.
- [13] Z. Yang et al., "Joint altitude, beamwidth, location, and bandwidth optimization for UAV-enabled communications," *IEEE Commun. Lett.*, vol. 22, no. 8, pp. 1716–1719, 2018.
- [14] NC. Luong et al., "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3133-3174, 2019.
- [15] A. Feriani et al., "Single and multi-agent deep reinforcement learning for AI-enabled wireless networks: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 1226-1252, 2021.
- [16] PS. Bithas et al., "A survey on machine-learning techniques for UAV-based communications," Sensors., vol. 19, no. 23, pp. 5170, 2019.
- [17] TP. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," *arXiv:1509.02971*, 2015.
- [18] H. Huang, Y. Yang, H. Wang, Z. Ding, H. Sari and F. Adachi, "Deep Reinforcement Learning for UAV Navigation Through Massive MIMO Technique," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 1117–1121, Jan. 2020, doi: 10.1109/TVT.2019.2952549.
- [19] O. Bouhamed, H. Ghazzai, H. Besbes and Y. Massoud, "Autonomous UAV Navigation: A DDPG-Based Deep Reinforcement Learning Approach," 2020 IEEE International Symposium on Circuits and Systems (ISCAS), Seville, Spain, 2020, pp. 1-5, doi: 10.1109/ISCAS45731.2020.9181245.
- [20] C. Wang, J. Wang, J. Wang and X. Zhang, "Deep-Reinforcement-Learning-Based Autonomous UAV Navigation With Sparse Rewards," in *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6180-6190, July 2020, doi: 10.1109/JIOT.2020.2973193.
- [21] M. Mahmood, M. Ghadaksaz, A. Koc and T. Le-Ngoc, "Deep Learning Meets Swarm Intelligence for UAV-Assisted IoT Coverage in Massive MIMO," in *IEEE Internet of Things Journal*, vol. 11, no. 5, pp. 7679-7696, 1 March1, 2024, doi: 10.1109/JIOT.2023.3318529
- [22] R. M´endez-Rial et al., "Hybrid MIMO architectures for millimeter wave communications: Phase shifters or switches?" *IEEE Access*, vol. 4, pp. 247–267, 2016.