IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 20 April 2023, accepted 30 April 2023, date of publication 22 May 2023, date of current version 7 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3278591

RESEARCH ARTICLE

A DRL Strategy for Optimal Resource Allocation Along With 3D Trajectory Dynamics in UAV-MEC Network

TAYYABA KHURSHID[1,2], WAQAS AHMED[1], MUHAMMAD REHAN[1], (Member, IEEE), RIZWAN AHMAD[3], MUHAMMAD MAHTAB ALAM[4], (Senior Member, IEEE), AND AYMAN RADWAN[5], (Senior Member, IEEE)

[1]Department of Electrical Engineering (DEE), Pakistan Institute of Engineering and Applied Sciences (PIEAS), Islamabad 45650, Pakistan
[2]Department of Telecommunication Engineering, Dawood University of Engineering and Technology (DUET), Karachi 74800, Pakistan
[3]School of Electrical Engineering and Computer Science, National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan
[4]Thomas Johann Seebeck Department of Electronics, Tallinn University of Technology, 19086 Tallinn, Estonia
[5]Department of Electronics, Telecommunications, and Informatics (DETI), University of Aveiro, and Instituto de Telecomunicações, 3810-193 Aveiro, Portugal

Corresponding author: Waqas Ahmed (waqas@pieas.edu.pk)

This work was supported in part by Telia Eesti AS through the Tallinn University of Technology (TalTech) Development Fund; in part by the Estonian Research Council under Grant PUT-PRG424 and Grant PUT-PRG667; in part by the European Union's Horizon Research and Innovation Program under Grant 668995, Grant 951867, and Grant 101058505; in part by the Tallinn University of Technology Development Program 2016–2022 under Grant 2014-2020.4.01.16-0032; and in part by the Fundacao para a Ciencia e a Tecnologia (FCT)/Ministério da Ciência, Tecnologia e Ensino Superior (MCTES) through national funds and when applicable co-funded European Union (EU) funds under Project UIDB/50008/2020-UIDP/50008/2020. The work of Tayyaba Khurshid was supported by the Faculty Development Program for Pakistani Universities (HEC).

ABSTRACT Advances in Unmanned Air Vehicle (UAV) technology have paved a way for numerous configurations and applications in communication systems. However, UAV dynamics play an important role in determining its effective use. In this article, while considering UAV dynamics, we evaluate the performance of a UAV equipped with a Mobile-Edge Computing (MEC) server that provides services to End-user Devices (EuDs). The EuDs due to their limited energy resources offload a portion of their computational task to nearby MEC-based UAV. To this end, we jointly optimize the computational cost and 3D UAV placement along with resource allocation subject to the network, communication, and environment constraints. A Deep Reinforcement Learning (DRL) technique based on a continuous action space approach, namely Deep Deterministic Policy Gradient (DDPG) is utilized. By exploiting DDPG, we propose an optimization strategy to obtain an optimal offloading policy in the presence of UAV dynamics, which is not considered in earlier studies. The proposed strategy can be classified into three cases namely; training through an ideal scenario, training through error dynamics, and training through extreme values. We compared the performance of these individual cases based on cost percentage and concluded that case II (training through error dynamics) achieves minimum cost i.e., 37.75 %, whereas case I and case III settles at 67.25% and 67.50% respectively. Numerical simulations are performed, and extensive results are obtained which shows that the advanced DDPG based algorithm along with error dynamic protocol is able to converge to near optimum. To validate the efficacy of the proposed algorithm, a comparison with state-of-the-art Deep Q-Network (DQN) is carried out, which shows that our algorithm has significant improvements.

INDEX TERMS MEC, offloading ratio, resource allocation, trajectory optimization, UAV dynamics.

I. INTRODUCTION

In the era of 5th Generation (5G) and Internet of Things (IoT), the need to analyze, process, and computation of huge chunks of data is becoming an interesting topic in the researchers' domain [1]. The complexity of executing computational tasks and data processing is becoming a challenge for End-user Devices (EuDs) due to their low storage, less computation power, and limited energy resources [2] and [3]. One solution to overcome the limitations of on-device local computing of EuDs is to use Multi-access Edge Computing (MEC) servers. MEC enables the EuDs to offload tasks to nearby

The associate editor coordinating the review of this manuscript and approving it for publication was Eyuphan Bulut.

edge servers, hence, reducing the extensive computational burden on EuDs [4]. MEC differs from conventional cloud computing in the sense that it uses Radio Access Networks (RANs) which are close to EuDs, resulting in low transmission delays [5]. Although MEC-based systems impose less burden on EuDs and utilize low communication and resources, there are several concerns due to which MEC-based systems are still in researchers' circle [6].

In earlier stages, MEC-based systems used fixed position Base Station (BS), and the ultimate goal was to enhance Quality of Service (QoS) and to reduce the computation burden on EuDs [7]. In [8], a comprehensive survey of mobile edge computing is provided, which focuses on service adoption and provision. The survey includes a detailed analysis of computational offloading as well as of deployment of edge-server and resource allocation. The question of how to place the edge servers for optimal performance is addressed in [9], where the placement of edge servers is formulated as a constrained optimization problem and Mixed Integer Programming (MIP) is applied to resolve the constrained problem. The computational offloading and resource allocation in a collaborative manner is studied for multi-layer MEC systems and vehicular MEC networks in [10] and [11] respectively.

Combining MEC networks with Reinforcement Learning (RL) manifests to be more efficacious because of the capacity of RL algorithms to work efficiently in highly nonlinear and dynamic environments, complex datasets, etc. [12]. A detailed survey about computation offloading strategies in MEC based on RL is presented in [13], where the authors compared RL strategies with supervised and unsupervised learning methods. The above survey also unleashed the open-ended issues and future prospects of integrating RL techniques in MEC networks. In [14], a DRL approach is used for joint task offloading and resource allocation, where cost, computation delay, and energy are minimized. The presented method is based on State-Action-Reward-State-Action (SARSA) algorithm to optimize resource management. A smart DRL-based resource allocation method is devised in [15] which allocates communication and computation resources adaptively by learning the environment and updating the policy.

A multi-user task offloading scenario is considered in [16], where offload is minimized and measured by energy consumption. The DDPG algorithm based on continuous action space is designed for decentralized computation offloading in [17]. The designed algorithm works without any prior knowledge of the network, and emphasis is given to a scenario where tasks arrive non-uniformly.

With the technological advances of UAVs, these devices are being used in many real-time applications like monitoring, remote sensing, security, surveillance, etc. [18]. A comprehensive survey on data collection in IoT networks by means of UAV is presented in [19]. Recently, UAVs are deployed for providing wireless coverage in scenarios where base stations are overloaded due to heavy traffic, communication facilities are sparsely distributed, and the occurrence of inevitable natural disasters or temporary malfunctions in BS is considered [20]. A recent interesting survey on advances in UAV-assisted wireless networks is presented in [21]. The survey is prominent from an optimization perspective and several optimization objectives are explored like delay, QoS, energy, coverage area, etc. In [22], the authors investigated resource allocation and UAV placement for IRS-assisted UAV-based wireless networks. The design focuses on maximization of the sum rate achieved by EuDs through optimizing UAV placement and IRS phase shift. The UAV placement problem is solved by leveraging from Successive Convex Approximation (SCA) method.

An optimization problem for total system delay is developed and Deep Q Network (DQN) is used to obtain the best resource allocation scheme [23]. The research study considered only a single UAV-edge server for providing auxiliary computation services to ground EuDs. To ensure the security of information and prevent eavesdropping in MEC networks, a full-duplex UAV is added to the MEC system to counter eavesdropping by sending interference signals [24]. To deal with the unbalanced traffic congestion on overloaded BS, a UAV network is used which integrates genetic algorithm and branch and bound method to optimize UAV position and spectrum efficiency respectively [25]. An artificial intelligence approach is used to elevate the energy efficiency of a UAV-based wireless network. The author compared the proposed AI strategy with federated deep learning (FDL) and multi-agent deep deterministic policy gradient (MADDPG) method [26].

It is worth mentioning that although literature on achieving UAV tracking performance is well developed, it is less useful in context of UAV based MEC network. As in UAV-MEC network, there are additional performance aspects that require optimization, such as delay, capacity, Age of Information, energy and etc. For interested readers', a summary of most relevant literature on UAV tracking performance is given below in the paragraph. An iterative learning control (ILC) design method was presented to improve tracking performance through learning from errors over iterations in repetitively operated systems [27]. A backstepping controller was introduced to improve tracking accuracy and robustness of UAVs' attitude control [28]. Conventional proportional and derivative lateral control law with some non-linear modifications were presented to enhance tracking performance for a UAV in different flight conditions [29]. Rapid transfer of controllers between UAVs using learning controllers was also proposed to improve trajectory tracking performance [30]. A comprehensive survey of control algorithms for UAVs was conducted, which covers many control and navigation techniques [31]. Proportional Integral Derivative (PID) controller, Linear Quadratic Regulator (LQR), Feedback Linearization Control (FLC), Linear Quadratic Gaussian (LQG), fuzzy logic, adaptive control, etc are some examples of UAV controllers [32].

TABLE 1. List of abbreviations.

Notation	Description
5G	Fifth Generation
AC	Actor Critic
AoI	Age of Information
BCD	Block Coordinate Descent
BS	Base Station
DCA	Difference Convex Algorithm
DDPG	Deep Deterministic Policy Gradient
DNN	Deep Neural Network
DQN	Deep Q Network
DRL	Deep Reinforcement Learning
EuD	End user Device
IMS	Improved Mean Shift
IoT	Internet of Things
IRS	Intelligent Reflecting Surfaces
MDP	Markov Decision Process
MEC	Multi-access Edge Computing
MINLP	Mixed Integer Non-Linear Programming
MIP	Mixed Integer Programming
MISO	Multi-Input Single-Output
QoS	Quality of Service
RAN	Radio Access Network
RL	Reinforcement Learning
SARSA	State-Action- Reward-State-Action
SCA	Successive Convex Optimization
UAV	Unmanned Aerial Vehicle

A. RELATED WORK

UAVs-assisted wireless networks offer numerous advantages in terms of coverage, mobility, cost, reconfiguration, and flexibility when compared to the deployment and operation of conventional wireless networks [33]. The UAVs equipped with wireless interfaces can act as a mobile base station to transmit (receive) data to (from) the network EuDs. The position and trajectory of these UAVs determine their coverage area and are strongly dependent upon the EuDs' density and traffic requirements [34]. However, a challenging situation to design an optimal scheme for the allocation of communication and computational resources along with the trajectory optimization exists because of UAV's limited onboard computation power, energy resources, and flight time.

In earlier works, the authors in [35] presented RL-based algorithm for optimal UAV positioning and transmission of power to drone small cells in order to revamp the outage performance and energy efficiency of UAVs. In [36], a UAV-assisted wireless sensor network is considered and authors developed a distributed RL strategy that permits devices to collaboratively update RL parameters. The objective was to minimize the weighted sum of Age of Information (AoI) cost in real-time and total energy consumption. A multi-UAV cooperative scenario is considered in [37] and a novel optimization algorithm for resource allocation and UAV positioning is proposed which can be split into two components: a) Deep Q Network approach is used to determine UAVs' position, b) Difference Convex Algorithm (DCA) is designed to work out UAV-EuD association and UAV beam-forming.

In recent works, a study is carried out on a multi-input single-output (MISO) UAV-based MEC network, aiming to optimize UAV's energy consumption, transmission power, and trajectory [38]. In [39], joint task assignment and UAV trajectory optimization problem are solved by using coalesced multi-population based genetic algorithm and dynamic programming. For efficient deployment and cost saving, an Improved Mean Shift (IMS) algorithm is presented in [40] to jointly optimize the number of UAV servers and their location. In [41], a Block Coordinate Descent (BCD) method is proposed to solve a non-convex optimization problem of reducing the overall energy consumption of EuDs with the local computing constraint and task completion deadline. For task scheduling and resource allocation, the branch and bound method is used, while for UAV trajectory optimization, SCA is employed. An interesting study regarding UAV speed optimization and path planning is witnessed in [42], where SCA and GA are manipulated to solve AoI and energy-aware-trajectory-optimization problem. A novel UAV-assisted IoT system is designed for the shortest UAV flight with maximum data collection from EuDs. A DRL technique is exploited to ensure maximum data collection with a significant sum rate while minimizing the flight path and usage of resources [43]. In [44], a cluster-based node mechanism is used which uses the k-value selection method, and UAV trajectories with minimum distance and total flight time are proposed. For an emergency scenario, optimization of UAV placement and trajectory planning for critical nodes is studied in [45]. Based on the capacity and Age of Information, two optimization problems are formulated and the RL technique is used to work out the optimal UAV placement. A recent resource allocation and 3D placement of the UAV-MEC network is studied in [46], where an iterative algorithm tends to jointly optimize UAV-EuD association, UAV's trajectory, task split ratio, and bandwidth allocation. Since the optimization incorporated the Mixed Integer Non-Linear Programming (MINLP) model, therefore SCA and BCD methods are applied to figure out the problem.

B. NOVELTY AND CONTRIBUTION

The existing literature summarized above often assumes a constant velocity model for the UAV. This leads to oversimplification, as the dynamics of UAVs are also equally important in determining the trajectory followed by the UAV. Even if an independent flight control model is assumed for trajectory tracking, the associated energy cost of the flight controller is ignored in the analysis. In addition, the cost factors are often not normalized [47] and [48], which simply lead to scale inconsistencies. Therefore, the following considerations have been covered in this paper:

- We propose a joint optimization approach of delay and energy, based on DRL for a UAV-MEC network with EuDs. The uncontrolled dynamics with input saturation (actions are bounded in a pre-specified region) are incorporated in DRL based model to efficiently offload tasks and manage resource allocation to achieve cost minimization.

- We propose a limited error feedback mechanism in DRL so that the error due to uncontrolled dynamics can be compensated.
- We further propose three schemes namely; training through an ideal scenario, training through error dynamics, and training through extreme value, and evaluate these to find the best scheme.
- We compare the performance of the above-mentioned schemes based on cost percentage, and conclude that case II (training through error dynamics) achieves minimum cost.

To authenticate the superiority of the propopsed DDPG based RL algorithm, simulations are performed and a comparison with the state-of-the-art DQN algorithm is performed. The comparison revealed that DDPG algorithm is able to give much btter optimal results.

The remaining paper is presented in sections as follows. In section II, a complete system model is defined, and an optimization problem is formulated. In section III, DDPG-based dynamic computation offloading and placement of UAV are discussed in detail. In section IV, simulation results and comparisons are provided to validate the effectiveness of the proposed strategy. The conclusion is provided in section V.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A multi-user UAV-MEC network is considered as illustrated in Figure 1. It consists of a UAV with an onboard MEC server and several EuDs $\mathcal{I} = \{1, 2, \ldots, I\}$. We consider a scenario where cellular network coverage becomes unavailable for EuDs (the EuDs cannot access resources from the base station). Due to the limited onboard capacity, and to ensure operation longevity, EuD offloads some portion of the task to the UAV. In this critical situation, UAV provides computational services to EUDs and executes a portion of the task, offloaded by EuD. A discrete-time model is used where the total time period $\mathcal{T}$ is divided into equal time slots [49] and only one EuD is served in each time slot t [50]. In subsequent subsections, the network model and computation model are introduced in detail.

A. NETWORK MODEL

We assume that there are $\mathcal{I}$ EuDs placed randomly in a pre-specified bounded area. A low-speed random mobility model has been assumed for all EuDs. The 3D location of EuDs is denoted by $x_i = (X_i, Y_i, H_i)$. In each time slot, the UAV has a starting point $x_j = (X_j, Y_j, H_j)$ and an end point $x_{j+1} = (X_{j+1}, Y_{j+1}, H_{j+1})$. The channel gain between UAV and EuD i can be written as

$$g_i(t) = \frac{c}{d_i^2(t)}, \tag{1}$$

where $d_i(t)$ denotes the distance between UAV and EuD i and c designates the channel gain at reference distance $d = 1$m. The uplink data rate between EuD i and UAV is calculated by

$$r_i(t) = W \log_2(1 + \frac{p_u g_i(t)}{\sigma^2 + b_i(t) P_{NLOS}}), \tag{2}$$

where W is the bandwidth, p_u is the transmission power of EuD, $g_i(t)$ is the channel gain, σ^2 is the noise power, $b_i(t) = \{0, 1\}$, 1 means blockage between UAV and EuD and 0 indicates no blockage, and P_{NLOS} is the transmission loss.

TABLE 2. Summary of notation.

Notation	Description
$\mathcal{I}$	Set of EuD devices
$\mathcal{T}$	Set of time slots
i	Single EuD device
$g_i(t)$	Channel gain of EuD i at time slot t
x_i	Location of EuD devices
x_j	Location of UAV
D	Task size
B	Number of CPU cycles requires for execution of task
p_u	Transmission power of EuD
f_i	Computation frequency of EuD device
f_j	Computation frequency of UAV
$b_i(t)$	Blockage between UAV and EuD at time slot t
W	Bandwidth of Network
$\mathbf{S}(t)$	State observed by UAV at time slot t
$\mathbf{A}(t)$	Action taken by UAV at time slot t
$\mathbf{R}(t)$	Reward obtained by UAV at time slot t
$\mathbf{S}(t+1)$	Transition state

FIGURE 1. Interconnection of EuDs in UAV-MEC Network.

B. COMPUTATION MODEL

We consider that each EuD has a computational task $\mathbb{R}$ to be completed in time period $\mathcal{T}$. In our scenario, the partial offloading technique is adopted by EuDs [51], and $\mathcal{R} = [0, 1]$ is the offloading ratio range between 0 and 1. Resultant, $1-\mathcal{R}$ is the remaining task to be executed locally by EuD i in time period $\mathcal{T}$.

1) LOCAL COMPUTATION MODEL

When EuD performs a computation task locally, it does not require any resource from the UAV or MEC server.

The local execution delay $\mathbf{T}$ of EuD i can be expressed as

$$\mathbf{T}_i^{loc}(t) = \frac{(1-\mathcal{R}_i(t))D_i(t)B}{f_i}, \tag{3}$$

where f_i is the computing capability of EuD, $D_i(t)$ denotes the task size of EuD i and B indicates the required CPU bits/cycle to process each chunk of data.

The energy consumed by EuD during the local execution of a task is calculated as

$$E_i^{loc}(t) = \mathcal{K}_i f_i^3 * \mathbf{T}_i^{loc}(t), \tag{4}$$

where $\mathcal{K}_i$ is the effective hardware switching capacity and f_i is the computational frequency of EuD.

2) OFFLOADING MODEL

When a portion of the EuD's task is executed by the UAV, the total delay experienced during the execution is the sum of uplink transmission delay, onboard execution delay, and downlink transmission delay. The downlink transmission delay is not considered here because the required data to be transmitted is too small [52]. The delay experienced during the processing of an offloaded task is divided into two parts. The first is a transmission delay and the second part is a computational delay. Transmission delay is represented by

$$\mathbf{T}_i^{tr}(t) = \frac{D_i(t)\mathcal{R}i(t)}{r_i}, \tag{5}$$

where r_i is the uplink rate between UAV and EuD. The computational delay experienced by EuD is denoted by

$$\mathbf{T}_i^{UAV}(t) = \frac{D_i(t)\mathcal{R}i(t)B}{f_{UAV}}, \tag{6}$$

where B is the required CPU bits/cycle and f_{UAV} is the computational capacity of UAV.

When EuD is associated with UAV for offloading its task, the energy consumption is calculated depending upon UAV constraints such as flying, hovering, and execution energy consumption at time slot t. At the end of each time slot, the UAV hovers from position x_j to the new position $x_{j+1}$with speed $v(t) \in [0, V_{max}]$ and angles $\theta = [0, 2\pi]$ and $\phi = [0, 2\pi]$. The energy consumption of UAV during this flight can be expressed as

$$E_{fly}^{UAV}(t) = p^{fly}||v(t)||^2, \tag{7}$$

where $p^{fly} = 0.5Mt_{fly}$, M is the mass of the UAV, t_{fly} is the flight time [52]. According to [53], the power consumption for task execution delay at time slot t is denoted by

$$P^{UAV}(t) = K_{UAV} f_{UAV}^3, \tag{8}$$

where K_{UAV} is the constant CPU cycle of UAV and f_{UAV} is the computing capacity.

The energy consumption of the UAV-MEC server is

$$E_{exe}^{UAV}(t) = P^{UAV}(t) * \mathbf{T}_i^{UAV}(t). \tag{9}$$

Then, the total energy utilization by the UAV is the sum of execution energy and flying energy and represented as

$$E_{UAV}(t) = E_{exe}^{UAV}(t) + E_{fly}^{UAV}(t). \tag{10}$$

C. PROBLEM FORMULATION

In this subsection, we formulate the optimization problem of minimizing the computational cost, which is the normalized function of processing time delay and energy consumption of UAV, by jointly optimizing 3D trajectory and resource allocation, subject to the network and computational model outlined above. The proposed work considers the UAV dynamics, which are often ignored in the existing literature. Newton's law of motion is followed to observe UAV dynamics and can be expressed as

$$dx_{UAV}(t) = v_{UAV}(t)dt. \tag{11}$$

where $v_{UAV}(t)$ is the speed of UAV at time t. Additionally, UAV adapt its velocity in accordance with current velocity. Also, UAV control factor has good effects on the velocity of UAV. Velocity dynamics of UAV can be expressed as:

$$dv_{UAV}(t) = (Av_{UAV}(t) + Bu_{UAV}(t))dt, \tag{12}$$

where $B = I$, $A = 0.1 * B$ and u_{UAV} is the input control variable [54]. The proposed aspect has a serious impact on UAV trajectory optimization and cost, which makes the optimization problem difficult to converge.

The aim is to minimize the computational cost considering the actual dynamics of UAV. In our model, we train the algorithm for 500 intervals. Each interval consists of 40 sets of time slots t and the computation cost is the mean of total energy and time delay in each interval of time $\mathcal{T}$. To add these two factors, we normalize energy and time delay values to calculate the total computational cost. For the notation, we say that the optimization variable (i.e., computational cost) is represented by z.

Mathematically, the optimization problem can be posed as follows.

$$\min_{\alpha_i(t), x_{j+1}, \mathcal{R}_i(t)} \mathbb{E}_{t=1}^{\mathcal{T}} \alpha_i(t)[(\mathbf{z}_i^{loc}(t) + \mathbf{z}_i^{off}(t))], \tag{13}$$

where

$$\mathbf{z}_i^{loc}(t) = \mathbf{T}_i^{loc}(t) + \mathbf{E}_i^{loc}(t),$$

and

$$\mathbf{z}_i^{off}(t) = \mathbf{T}_i^{tr}(t) + \mathbf{E}_i^{tr}(t) + \mathbf{T}_i^{UAV}(t) + \mathbf{E}_i^{UAV}(t).$$

When EuD is associated with UAV, $\alpha = 1$ otherwise $\alpha = 0$. The above optimization problem is to be solved subject to the following network, computation, and environmental constraints.

$$C1 : \mathcal{R}_i(t) \in \{0, 1\}, \forall t, i,$$

$$C2 : \sum_{I=1}^{\mathcal{I}} \alpha_i(t) = 1, \forall t,$$

$$C3 : x_j(t) = X_{min} \leq x_j \leq X_{max},$$

$$Y_{min} \leq y_j \leq Y_{max},$$

$$H_{min} \leq h_j \leq H_{max}$$

$$C4 : b_i(t) \in [0, 1], \forall t, i,$$

$$C5: \sum_{t=1}^{\mathcal{T}} \sum_{I=1}^{\mathcal{I}} \alpha_i(t) D_i(t) \leq D,$$
$$C6: \sum_{t=1}^{\mathcal{T}} E_{UAV}(t) \leq E_b, \forall i,$$

where $C1$ indicates that offloading ratio ranges between 0 and 1. $C2$ expresses that UAV serve at a maximum of one EuD in each time slot. $C3$ shows that UAV can move only in the specified area. $C4$ represents the blockage between UAV and EuD. $C5$ ensures that all computing tasks are completed in a pre-defined time period $\mathcal{T}$. $C6$ means that the total energy consumed by UAV does not exceed the maximum battery capacity of UAV.

III. DRL BASED COST OPTIMIZATION AND PLACEMENT OF UAV WITH DYNAMICS

Deep reinforcement learning (DRL) is a variant of reinforcement learning that involves deep neural networks to approximate the Q-value function or policy function in RL. This allows for more complex and sophisticated decision-making by the agent, as the neural network can learn to represent complex state-action mappings. This paper involves complex state action mapping as indicated by the system Model in subsection III-B.

In general, Deep reinforcement learning has several benefits over conventional and simple reinforcement learning methods. Deep RL algorithms can achieve great performance on complex tasks [55], [56] even without the need for prior knowledge about the environment [57]. In this paper, although, the dynamic model of the UAV is known, the dynamic model of delay and energy for mobile EuDs in terms of UAV trajectory is not known. The problem in this paper focuses on the optimization of combined energy and delay by achieving a better approximation of Q-value function using deep network, which defines the probability of actions taken by the UAV. As the model is not completely known, the deep network develops a policy function approximation based on Q-value approximation through a neural network that can provide the maximum reward.

Many DRL approaches are proposed in the literature such as Deep Q-Network (DQN) [58], Deep Deterministic policy gradient (DDPG) [59], Deep State Action Reward State Action (Deep SARSA) [60], [61], and Double DQN [62]. These algorithms are employed in UAV control to achieve superior performance. However, all these works do not consider the dynamics of the UAV. In general, the DDPG approach is more suited to continuous action space, whereas DQN approaches are more suited for discrete action space. Deep SARSA is a more simple on policy approach that relies on DQN based Q-value evaluation. Since, we are assuming a continuous time dynamic model for the UAV, therefore, to obtain a better policy function approximation, we consider a continuous action space and propose a DDPG-based UAV cost optimization.

DDPG is a modern reinforcement learning system that approximates the Q-value action function using two neural networks, a critic network that generates unique actions using an actor network. The DDPG algorithm is used to determine the best action for UAV-assisted MEC system's user scheduling, UAV mobility, and resource allocation. We present a Deep Deterministic Policy Gradient (DDPG) based offloading method, which successfully supports a continuous action space, and provides flexibility of training and tuning two neural networks, i.e., actor network and critic network. Actor network maximizes the Q-value estimated by the critic network which in turn corresponds to higher expected rewards.

Our solution includes UAV dynamics and minimizes the computational cost (i.e., the normalized function of processing time delay and energy consumption of UAV). In a practical environment, we observe that the UAV movement is not proportional to the output generated by the system. Moreover, if DDPG and DQN are unaware of the uncontrolled UAV dynamics, the resultant action may not lead to the desired result. This results in inadequate learning and requires a dynamic controller.

In order to accommodate the uncontrolled dynamics, we have also incorporated a mechanism which tends the UAV to learn the error accumulated during the trajectory. When these learning techniques are applied to a UAV-MEC RL framework, it learns the computation offloading policy and selects an action, i.e., EuD to be served, offloading ratio, and placement of UAV. Now, we explain the state, action, and reward function of the UAV-MEC system.

Agent: UAV is an independent agent of the RL environment that learns an optimal policy to maximize its reward in each time step $\mathcal{T}$. UAV learns policy, executes an action, and based on that action a reward is generated. UAV is able to move in a constrained environment composed of EuDs along with some parametric limitations such as height, flying time, etc.

EuDs: EuDs are Edge user Devices that connect to UAV in absence of BS to offload some segment of the task. EuDs adopt a mobility model that allows them to move in a pre-specified environment at a very low speed.

States: The state space can be represented as $\mathbf{S} = \{E_b, D, \mathbb{R}, x_i, x_j, b_i\}$, where E_b is the battery capacity of UAV, D is the sum task size, $\mathbb{R}$ is the task size information of each EuD, x_i and x_j are the location of EuDs and UAV respectively, and b_i is the blockage between UAV and EuDs. It is the set of all possible states for the UAV. We apply state normalization to reduce the distinction between the magnitude of different states by taking the difference between maximum and minimum values of state and using it as a scaling factor. The states are determined based on the constraints of EuDs, UAV, and the environment.

Actions: We define the following actions based on the current state, environment, UAV dynamics, and system constraints. UAV can take the following three actions.

1) EuD association $\mathcal{A}$: UAV selects EuD to be served in time slot t. UAV offers its services to each EuD but only one EuD in a unit time slot.
2) Offloading ratio $\mathcal{R}$: UAV sets the portion of the task to be offloaded by the EuD. Offloading ratio ranges between 0 and 1. 1 means EuD offloads its full task to UAV. As the ratio decreases, the percentage of offloading tasks also reduces.
3) Position $\mathcal{L}$: UAV selects next position by obtaining distance, ϕ (longitude) and θ (latitude) value as action.

Action space is denoted by $\mathbb{A} = \{\mathcal{A}, \mathcal{R}, \mathcal{L}\}$.

Reward: The reward is the objective function represented as equation (13). When EuD offloads its task to UAV, EuD experiences some delay in the transmission and execution of tasks. Moreover, there is a bounded energy constraint on both EuDs and UAV. Our objective is to minimize the mean normalized computational cost while maximizing the reward. The Reward function includes the total computational cost in executing the task by EuD and can be written as:

$$\mathbf{R}(t) = -z_{total}, \tag{14}$$

where z_{total} is the normalized computational cost and equal to the mean of processing delay and energy consumption in time period $\mathcal{T}$.

A. MDP MODEL

A Markov decision process (MDP) consists of a 4-tuple $< \mathbf{S}, \mathbf{A}, \mathbf{R}, \mathbf{P} >$ where $\mathbf{S}$ is the state space, $\mathbf{A}$ is the action space, $\mathbf{R}$ is the expected reward and $\mathbf{P}$ is the transition probability from the current state to the next state [63]. In each time slot t, the environment is in state $\mathbf{S}_t$, UAV observes the current state and selects action $\mathbf{A}_t$ according to current policy π. The environment grants the UAV with reward $\mathbf{R}_t$ (normalized cost value) and transits into the next state $\mathbf{S}_{t+1}$ as per transition probability of environment $p(\mathbf{S}_{t+1}|\mathbf{S}_t, \mathbf{A}_t)$ [17]. MDP goals are to determine the optimal policy that maximizes the expected collective reward as

$$R_t = \sum_{l=t}^{T} \gamma^{l-t} \mathbf{R}_{(\mathbf{S}_l, \mathbf{A}_l)}, \tag{15}$$

where $\gamma \in [0, 1]$ represents the discount factor and $\mathbf{R}_{(\mathbf{S}_l, \mathbf{A}_l)}$ is the instant reward at t^{th} time slot. Under policy π, the expected discount return from state $\mathbf{S}_t$ is defined as the state value function.

$$V^{\pi}(\mathbf{S}_t, \mathbf{A}_t) = \mathbb{E}_{\pi}[R_t|\mathbf{S}_t]. \tag{16}$$

The state action function is the expected discounted return after taking action $\mathbf{A}_t$ in state $\mathbf{S}_t$ under a policy π, i.e.,

$$\mathcal{Q}^{\pi}(\mathbf{S}_t, \mathbf{A}_t) = \mathbb{E}_{\pi}[\mathbf{R}_t|\mathbf{S}_t, \mathbf{A}_t]. \tag{17}$$

The basic property of MDP is the Bellmen equation that represents the iterative relationship between the state-value function and action-value function as

$$V^{\pi}(\mathbf{S}_t, \mathbf{A}_t) = \mathbb{E}_{\pi}[\mathbf{R}(\mathbf{S}_t, \mathbf{A}_t) + \gamma V^{\pi}(\mathcal{S}_{t+1})], \tag{18}$$

$$\mathcal{Q}^{\pi}(\mathbf{S}_t, \mathbf{A}_t) = \mathbb{E}_{\pi}[\mathbf{R}(\mathbf{S}_t, \mathbf{A}_t) + \gamma \mathcal{Q}^{\pi}(\mathbf{S}_{t+1}, \mathbf{A}_{t+1})]. \tag{19}$$

B. DDPG

To deal with extensive state and action space issues, we present DDPG to optimize the normalized computational cost of UAV. Despite the fact that DQN effectively solved issues in high-dimensional state spaces but continuous action spaces are still difficult to handle by DQN [64]. DDPG is proposed to expand DRL algorithms to continuous action spaces [65]. As shown in Figure 12, the presented DDPG model includes two networks, an actor network and a critic network. Actor network of UAV takes observations $\mathbf{S_t} = \{E_b, D, \mathbb{R}, x_i, x_j, b_i\}$ and provides action $\mathbf{A_t} = \{\mathcal{A}, \mathcal{R}, \mathcal{L}\}$ for the current state. Based on the actions, UAV serves its resources to EuD. The network state enters into the next state S_{t+1} and gains some reward for the UAV. Actor-Network stores tuple $(\mathbf{S}_t, \mathbf{A}_t, \mathbf{R}_t, \mathbf{S}_{t+1})$ in replay memory buffer. Critic network takes observation $\mathbf{S_t}$ and corresponding action $\mathbf{A_t}$ and gives Q value as output in each time slot t. Q value indicates how beneficial was the action we took and improve it in the next time slot. Q value is equal to the reward for the current action plus discounted Q next as

$$\mathcal{Q} = [(\mathbf{R} + \gamma \mathcal{Q}(\mathbf{S}', \mathbf{A}|\theta')].$$

The critic target network determines the target Q-value for training the critic-main network as

$$y_t = (\mathbf{R}_t + \gamma \max_{a \in \mathcal{A}} \mathcal{Q}(\mathbf{S}', \mathbf{A}|\theta')).$$

The critic target network sends y_t to the critic main network to minimize the loss function.

$$L(\theta)^{\mathcal{Q}} = \mathbb{E}_{\mu'}[y_t - \mathcal{Q}(\mathbf{S}, \mathbf{A}|\theta)^2]. \tag{20}$$

Actor network takes random sample states from memory (memory has all records of states, actions, and rewards but we take only states) and determines actions for those states. These actions may be different from the action we stored in memory B. Send these actions from the actor network into the critic network along with the states and get the value for the critic network. Now, the actor network takes the gradient of the critic network with respect to the parameter of actor network. The actor network updates itself according to [66]

$$\nabla_{\theta^{\mu}} J = \mathbb{E}_{\mu'}[\nabla_a \mathcal{Q}(\mathbf{S}, \mathbf{A}|\theta^{\mathcal{Q}}) \nabla_{\theta^{\mu}} \mu(\mathbf{S}|\theta^{\mu})]. \tag{21}$$

In each time slot t, a soft update rule is used $\theta^{\mathcal{Q}'}$ and $\theta^{\mu'}$. Update those with τ multiply by the value of the online network and add in the current value of the target actor or target critic network.

From equations (20) and (21), the actor and critic network parameters can be updated by $\theta^{\mathcal{Q}'} \leftarrow \theta^{\mathcal{Q}} - \alpha^{\mathcal{Q}}$ and $\theta^{\mu'} \leftarrow \theta^{\mu} - \alpha^{\mu}$.

C. DQN

DQN technology uses a parameterized DNN to approximate the Q-values $\mathcal{Q}(\mathbf{S}, \mathbf{A})$ [67]. To solve the problem of instability while using the function approximation in RL, the UAV initializes a replay memory buffer by executing completely random actions for a few time steps. Then UAV makes two

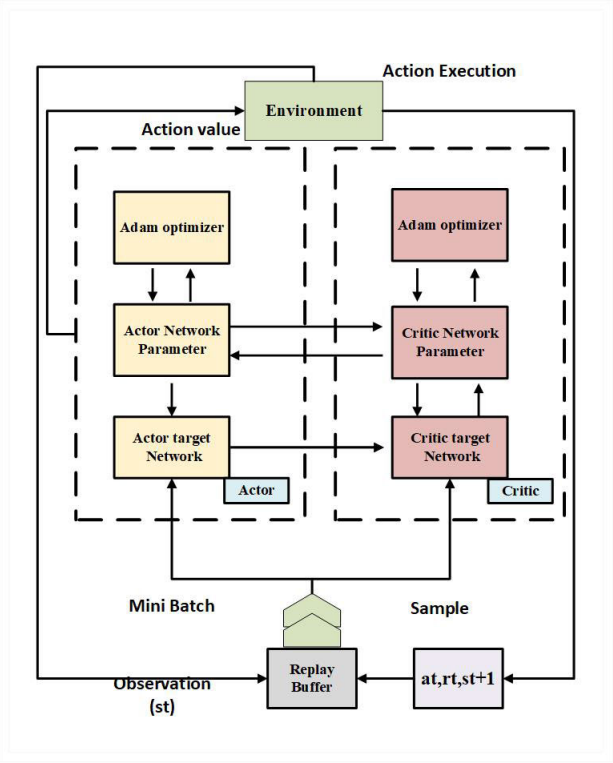

FIGURE 2. Block diagram of DDPG.

cases of DQN network, i.e., the online network and the target network. UAV adjusts the weights of the target network identically to the online network. At each time slot t, UAV uses an epsilon greedy strategy to decide whether to execute random action or perform the action as per the network.

$$a_t = \begin{cases} \text{random action } \mathbf{A_t} & \text{with probability } \epsilon \\ arg\,max_{a\in\mathcal{A}}\mathcal{Q}(s_t, a|t) & \text{otherwise} \end{cases} \quad (22)$$

UAV executes the selected action $\mathbf{A_t}$ to obtain the immediate reward $\mathbf{R}$ and next state $\mathbf{S_{t+1}}$ and store this experience $\mathbf{S}_t, \mathbf{A}_t, \mathbf{R}_t, \mathbf{S_{t+1}}$ to replay memory buffer B, the UAV then draws a mini-batch of random samples from the memory and computes the target Q value using the target network. UAV computes predicted Q value using an online network. Loss between the targeted and the predicted Q value is calculated by UAV to update the weights of the online network. At regular intervals, UAV makes a copy of the weights of the online network into the target network.

The target value of UAV is upgraded slowly but the main Q-value is upgraded frequently. Thus, the correlation between the target value and Q-value decreases that makes the algorithm stable. In each time slot, the deep Q-function is trained by minimizing the loss function $L(\theta)$ which is given as

$$L(\theta) = \mathbb{E}[(\mathbf{R} + \gamma \max_{\mathbf{A}\in\mathcal{A}} \mathcal{Q}(\mathbf{S}', \mathbf{A}|\theta') - \mathcal{Q}(\mathbf{S}, \mathbf{A}|\theta)^2], \quad (23)$$

where $(\mathbf{R}+\gamma \max_{\mathbf{A}\in\mathcal{A}} \mathcal{Q}(\mathbf{S}', \mathbf{A}|\theta'))$ denotes the target value of the network and $\mathcal{Q}(\mathbf{S}, \mathbf{A}|\theta)$ is the Q-value. Loss function is used

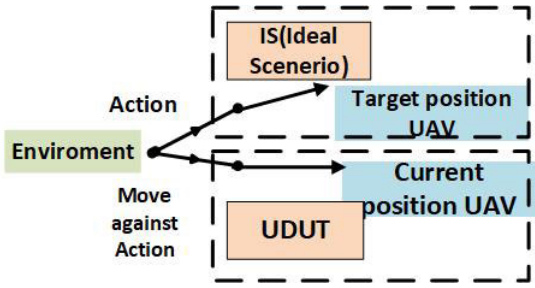

FIGURE 3. Virtual diagram.

to update the network parameter by $\theta \leftarrow \theta - \alpha.\nabla_\theta L(\theta)$ with a learning rate α.

D. TRAINING AND TESTING

To realize DDPG based computation offloading strategy, training and testing phase are the two phases of the DRL framework. The training process is illustrated in Algorithm 1. In each time slot t, UAV starts with initial state $\mathbf{S}_{t,1}$ and terminates at maximum step T. UAV learns tuple $(\mathbf{S}_t, \mathbf{A}_t, \mathbf{R}_t, \mathbf{S}_{t+1})$ and stores in replay memory buffer B. Meantime, UAV actor and critic network are updated using mini-batch tuples that are randomly selected from replay memory buffer B. Thus, after training the maximum length of episode T, the UAV is able to learn to optimize computation offloading and UAV placement policy. For the testing phase, UAV first gets its learned parameter of the actor from the training phase. Then, UAV initializes an empty data buffer B and a random environment is considered. Afterward, the current state is sensed by UAV and the corresponding action is selected according to the output of the actor network. We consider three different scenarios in this paper and compare the results.

1) CASE 1 (IS)

In this case, an Ideal Scenario (IS) of resource allocation is considered, where the computational cost is observed in the absence of UAV dynamics. This is equivalent to the scenario considered in earlier works [47] and [68] and similar to DDPG. In each time slot t, UAV starts with initial state $\mathbf{S}_t$ and learns tuple $(\mathbf{S}_t, \mathbf{A}_t, \mathbf{R}_t, \mathbf{S}_{t+1})$ and store it in buffer memory B. UAV senses the current state and gives action, reward, and next state value as output. UAV then executes the action and trains the critic network by minimizing the error between $\mathcal{Q}$ and $y(t) = (\mathbf{R} + \gamma\mathcal{Q}')$. It trains the actor network by maximizing $\mathcal{Q}$ using deterministic policy gradients. After this, network parameter soft replacement is updated and performs trajectory. This is an ideal scenario of resource allocation of UAV for comparison of proposed work.

2) CASE 2 (UDUT)

In this case, Uncontrolled Dynamics of UAV trajectory (UDUT) are considered to observe how the dynamics of UAV affect the value of cost. We include these dynamics in our

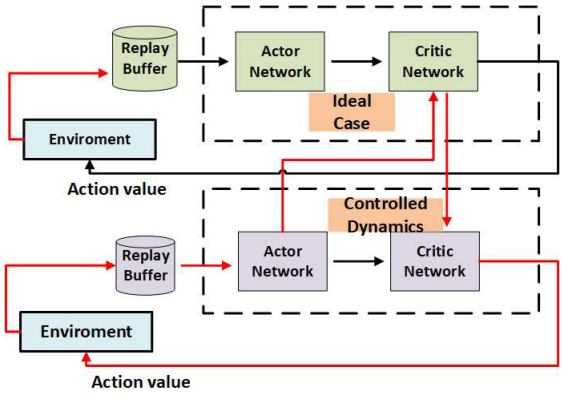

FIGURE 4. Virtual training network.

model because UAV does not follow the trajectory required by the DDPG and DQN to move in a real environment. The virtual Diagram for this case is shown in Figure 3. We assume that the actor and critic are unaware of error dynamics. At the start of the algorithm, UAV initializes its learning parameter of actor and critic and considers an empty data buffer B. Actor Network considers tuples stored in buffer memory and takes action against states. We pass that action through a system dynamic equation and observe how actions change as we add dynamics to our system. Critic observe how good was the action and trains Q function by minimizing the loss function. The loss function updates the network parameter with learning rate α.

3) CASE 3 (CDUT)

In this case, Control trajectory to adjust Dynamics (CDUT) is performed in presence of UAV dynamics and the model is trained to reduce the difference between the actual position and desired position of the UAV. The virtual training network of this case is shown in Figure 4. We carry out three cases to minimize error and select the best case for our comparison with IS and UDUT. Three cases are as follows:

a: CDUT (TRAINING THROUGH IDEAL SCENARIO)

In this sub-case, UAV dynamics are observed and the model is trained to achieve the target value of the critic network of an IS. For this purpose, we consider two actor networks and two critic networks. Each one is for IS and CDUT. For both environments, UAV set its learning parameter from the training phase and initializes an empty data buffer. Afterward, the current state is observed by the UAV and the corresponding action is selected. Actor takes observation from memory and provides action. Now, Critic network of UDUT receives an action value to calculate Q and receives a target value from IS critic network as

$$y_t = \mathbf{R}_t + \gamma \max_{\mathbf{A}\in\mathcal{A}} \mathcal{Q}'_{IS}. \tag{24}$$

Critic network CDUT minimizes the difference between the ideal case target value y_t and achieved main Q value.

b: CDUT (TRAINING THROUGH ERROR DYNAMICS)

In this case, UAV calculates the error between the desired trajectory and the current trajectory through the following equation.

$$(X_d - x(t)/V^x_{max}) + (Y_d - y(t)/V^y_{max}) + (H_d - h(t)/V^h_{max})$$

where $\{X_d, Y_d, H_d\}$ is the desired UAV position and $\{x(t), y(t), h(t)\}$, is the current position of UAV. UAV sends this error value to the critic target network to minimize the variation between the original position and the target position. Critic Network adds this error value in the target network as

$$y_t = \mathbf{R}_t + \gamma \max_{\mathbf{A}\in\mathcal{A}} \mathcal{Q}' + error, \tag{25}$$

and train itself to minimize the difference between $\mathcal{Q}(\mathbf{S}_t, \mathbf{A}_t|\theta)$ main network value and target network y_t value.

c: CDUT (TRAINING THROUGH EXTREME VALUE)

In this sub-case, UAV dynamics are observed and trained on the target value that is the maximum of IS case and UDUT case. In this case, two actor and critic networks are considered as in CDUT (Training through IS). Actor network takes action based on observations and passes that action to the critic network. Critic network receives action value and takes the maximum value of the target network of IS case and UDUT case as its target value.

$$y_t = \mathbf{R}_t + \gamma max\{\max_{\mathbf{A}\in\mathcal{A}} \mathcal{Q}'_{IS}, \max_{\mathbf{A}\in\mathcal{A}} \mathcal{Q}'_{UDUT}\}. \tag{26}$$

Then, minimize the difference between the target value y_t and achieve the main Q value. Now, the actor network takes the gradient of the critic network and updates itself and updates itself accordingly.

IV. NUMERICAL EXPERIMENT

For the simulation purpose, we make use of Python v3.7.0 along with TensorFlow library v1.14.0. Also, we utilized MATLAB ode45 function library to deal with uncontrollable UAV dynamics. In the subsequent section, we introduce all the parameters used for the simulation.

A. SIMULATION SETUP

In our UAV-MEC system, there is one independent UAV equipped with MEC server and a multi-user environment is considered where four EuDs are deployed. We categorize these parameters into three types namely; network and RL parameters, EuDs parameters, and UAV parameters.

1) NETWORK AND RL PARAMETERS

The line-of-sight noise power and non-line-of-sight noise power are taken as -100dBm [52] and -80dBm [69] respectively. The reference channel gain at a distance of 1m is -50dBm [52]. The time period $\mathcal{T}$ is 320 seconds and divided into 40 slots of 8 seconds each. The number of CPU cycles required to process one data unit is 1000 cycles [52]. The learning rate of the actor and critic network is 0.001 and 0.002 respectively. Other RL parameters are discount

Algorithm 1 DDPG Algorithm With UAV Dynamics

Initialize the actor network and critic network with weight θ^{μ} and $\theta^{\mathcal{Q}}$ and an empty replay memory buffer B_t.
Set target network parameters $\theta^{\mu} \leftarrow \theta^{\mu'}$ and $\theta^{\mathcal{Q}} \leftarrow \theta^{\mathcal{Q}'}$
for episodes = 1, N **do**
  Reset Simulation parameters and get initial state s_1.
  for t = 1,2,$\cdots$ $\mathcal{T}$ **do**
    Select action a'.
    if case = IS **then**
      a is equal to a'.
      execute action a.
      Observe next state s' and get reward **R**.
      Store (s, a, r, s') in replay memory buffer.
      Randomly sample mini batch of transition B.
      Compute target
      $y_t = \mathbf{R}_t + \gamma \max_{a \in \mathcal{A}} \mathcal{Q}(s', a|\theta')$.
    else if case = UDUT **then**
      Pass the selected action a' through ODE45 function to consider UAV dynamics.
      Adopt output value of ODE45 function as final control action a.
      execute action a.
      Observe next state s' and get reward **R**.
      Store (s, a, r, s') in replay memory buffer.
      Randomly sample mini batch of transition B.
      Compute target
      $y_t = \mathbf{R}_t + \gamma \max_{a \in \mathcal{A}} \mathcal{Q}(s', a|\theta')$.
    else if case = CDUT **then**
      Pass the selected action a' through ODE45 function to compensate dynamics.
      Adopt output value of ODE45 function.
      calculate error e between desired trajectory and current trajectory.
      execute action a.
      Observe next state s' and get reward **R**.
      Store (s, a, r, s') in replay memory buffer.
      Randomly sample mini batch of transition B.
      Compute target
      $y_t = \mathbf{R}_t + \gamma \max_{a \in \mathcal{A}} \mathcal{Q}(s', a|\theta') + error$.
    end if
    Update $\theta^{\mathcal{Q}}$ in critic network by minimizing loss function $L(\theta)^{\mathcal{Q}} = \mathbb{E}_{\mu'}[y_t - \mathcal{Q}(s, a|\theta)^2]$.
    Update $\theta^{\mu'}$ critic network by sampled policy gradient
    $\nabla_{\theta^{\mu}} J = \mathbb{E}_{\mu'}[\nabla_a \mathcal{Q}(s, a|\theta^{\mathcal{Q}}) \nabla_{\theta^{\mu}} \mu(s|\theta^{\mu})]$.
    Update target networks by
    $\theta^{\mathcal{Q}'} \leftarrow \tau\theta^{\mathcal{Q}} + (1-\tau)\theta^{\mathcal{Q}'}$
    $\theta^{\mu'} \leftarrow \theta^{\mu} + (1-\tau)\theta^{\mu'}$.
  end for
end for

factor = 0.001, exploration rate = 0.01, and soft update factor = 0.01.

2) SIMULATION PARAMETERS

The uplink power of EuDs is taken as 0.1 W [70]. The limited Computation frequency of EuD and UAV is 0.6 GHz and 1.2 GHz respectively [52]. The task size of EuD varies from 2-2.5 Mbits. The constant mass of UAV is 9.65 Kg and battery of UAV is 500 KJ [71]. The maximum speed at which UAV can fly is 15 m/s [72]. The time of flight and time of hovering is 1 sec and 7 sec respectively. The bandwidth is selected as 1 MHZ [73].

The detailed simulation parameters are listed in Table 3.

TABLE 3. Simulation parameters.

Parameter Type	Value
Number of UAV	1
Number of BS	1
Number of EuDs (I)	4
Uplink Power of EuD (P_u)	0.1 W
Noise power LOS (P_{LOS})	-100 dBm
Noise power NLOS (P_{NLOS})	-80dBm
Reference channel gain at distance of 1m (α_o)	-50dB
Number of slots in one Time period (t)	40
UAV mass/kg (M_{UAV})	9.65
Computation Frequency of UAV (f_{UAV})	1.2 GHz
Computation Frequency of EuD (f_{EuD})	0.6 GHz
Task size of EuD (D)	[2-2.5] Mbits
Number of CPU cycles required for unit bit processing(B)	1000 cycles/bit
Time of flight (t_{fly})	1 sec
Time of hovering (t_{hov})	7 sec
Bandwidth (W)	1 MHz
Learning rate (α_{Actor}) (α_{Critic})	 0.001 0.002
Discount factor(γ_{factor})	0.001
Exploration rate (σ_e)	0.01
Soft update factor (τ)	0.01

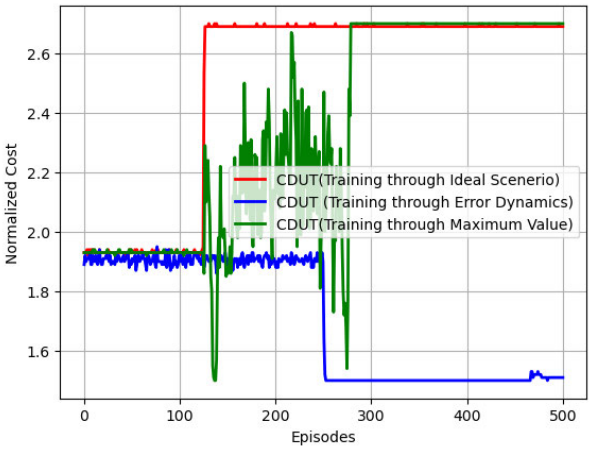

FIGURE 5. All error cases of DDPG per episode.

B. RESULTS AND DISCUSSION

In this subsection, we illustrate the results obtained by manipulating the proposed DDPG based algorithm. For this purpose, we consider three cases; a) ideal scenario (IS) where UAV dynamics are ignored, b) uncontrolled dynamics of UAV trajectory (UDUT) which considers UAV dynamics and c) controlled dynamics of UAV trajectory (CDUT) where UAV dynamics with addition of compensating factor is used.

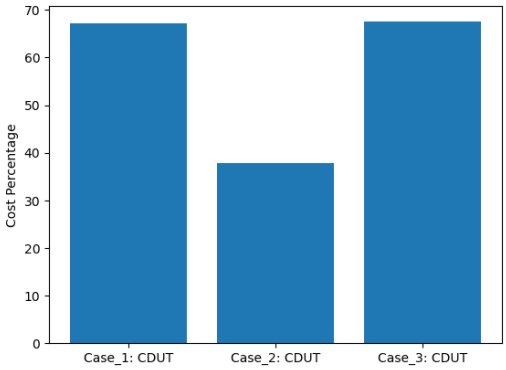

FIGURE 6. All error cases of DDPG per episode.

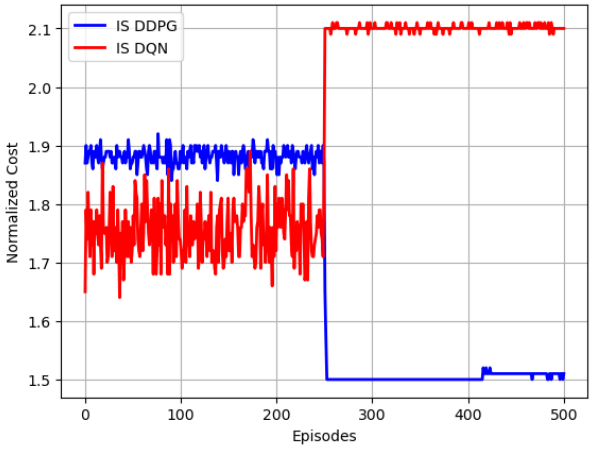

FIGURE 7. CASE 1 -average cost of DQN and DDPG per episode.

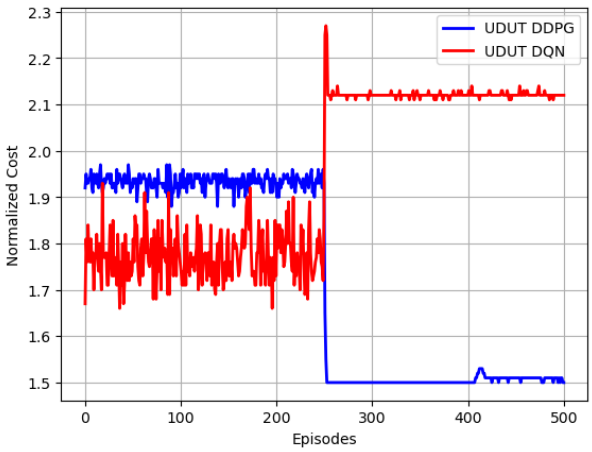

FIGURE 8. CASE 2 -average cost of DQN and DDPG per episode.

We further divide CDUT into three sub-cases i.e., CDUT (training through Ideal Scenario), CDUT (training through error dynamics), and CDUT (training through maximum Q value), and compare the performance of all three sub-cases. Figure 5 shows that CDUT (training through error dynamics) achieves minimum cost value, i.e., 1.51.

Figure 6 shows the percentage cost of all three cases. It can be seen that CDUT (training through error dynamics) has

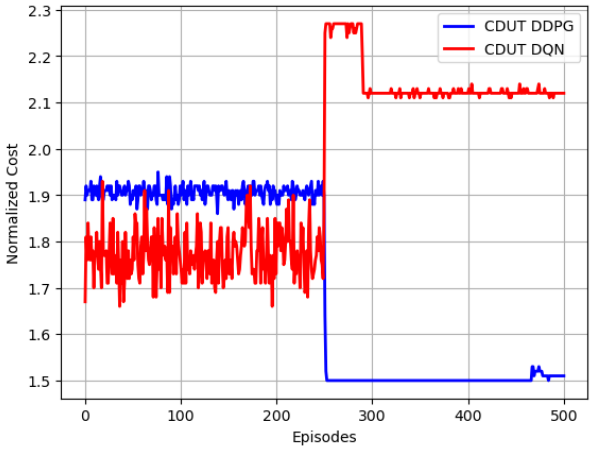

FIGURE 9. CASE 3 -average cost of DQN and DDPG per episode.

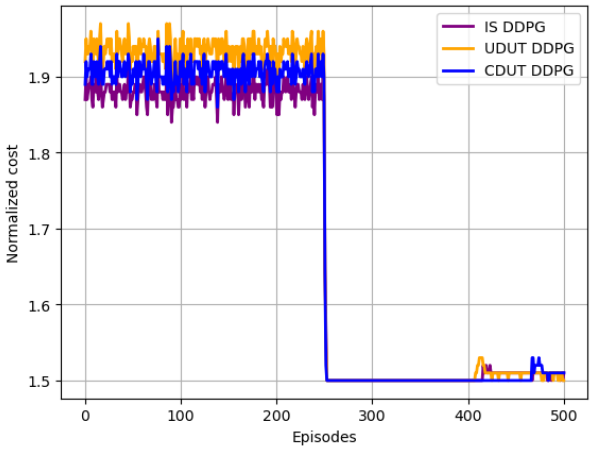

FIGURE 10. Average cost of DDPG per episode.

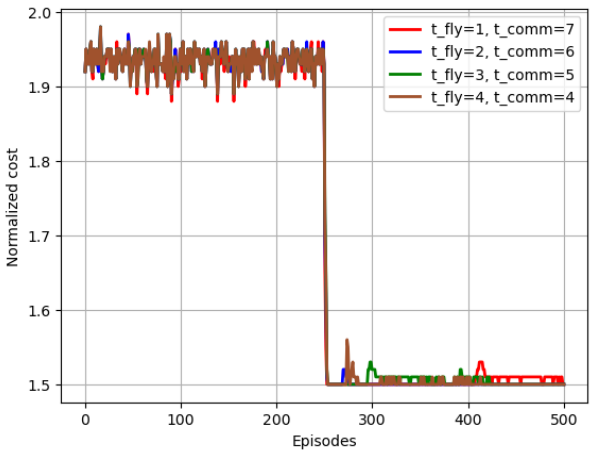

FIGURE 11. Average cost of DDPG vs different flying time per Episode.

minimum cost value as compared to CDUT (training through Ideal Scenario) and CDUT (training through maximum Q value). Therefore, we consider CDUT (training through error dynamics) in our system model. We observe all three cases in terms of DDPG and DQN. Figure 7 shows the average

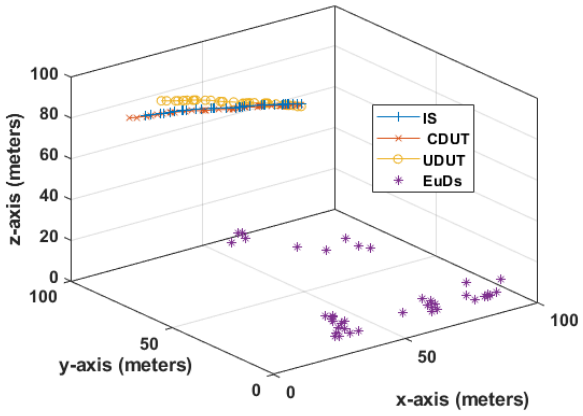

FIGURE 12. 3D view of UAV trajectory.

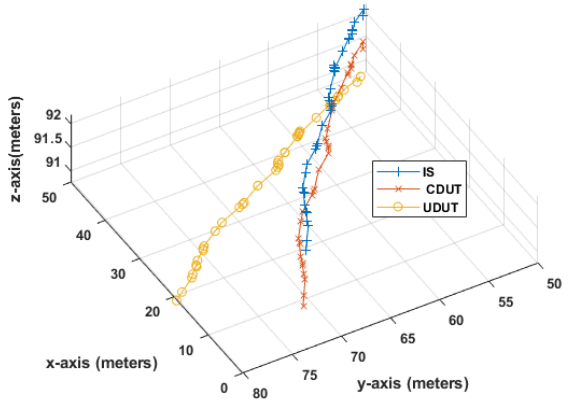

FIGURE 13. Close view of UAV trajectory.

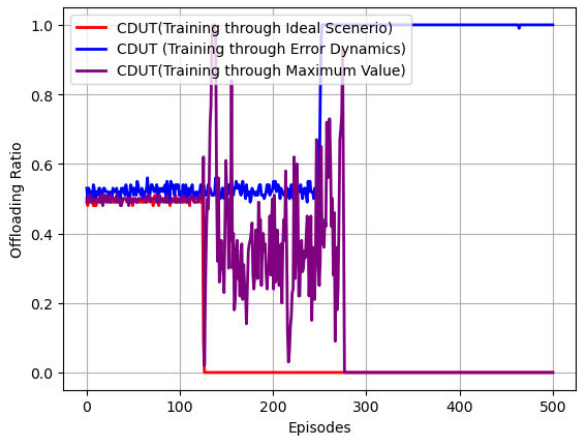

FIGURE 14. CDUT offloading ratio per Episode.

cost per episode of DDPG and DQN technique for IS. Figure 8 shows the average cost per episode of DDPG and DQN technique for UDUT case. As we can see, the cost is effected and increased as we added dynamics factor in our system. To minimize the difference between IS and UDUT, we added an error factor in our system as shown in Figure 9.

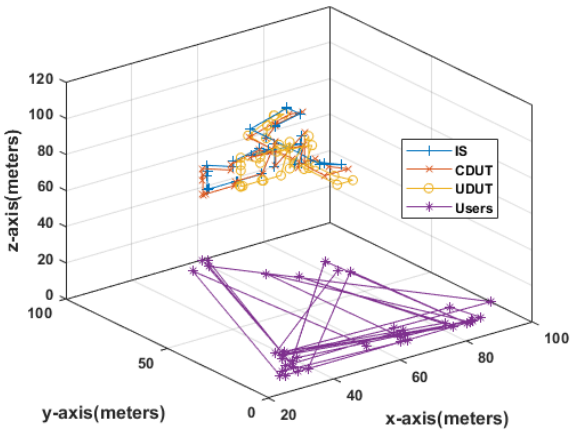

FIGURE 15. UAV trajectory.

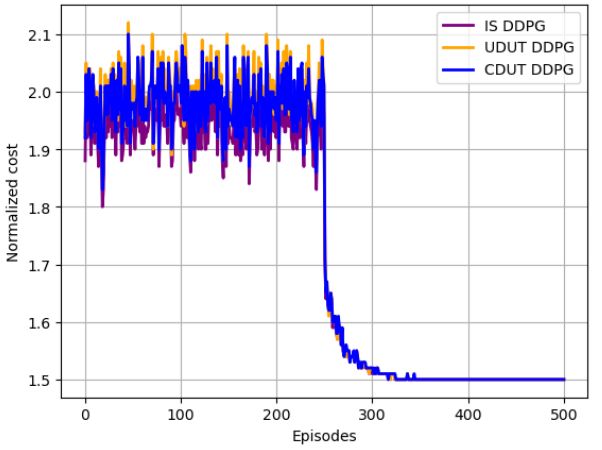

FIGURE 16. Optimized cost DDPG.

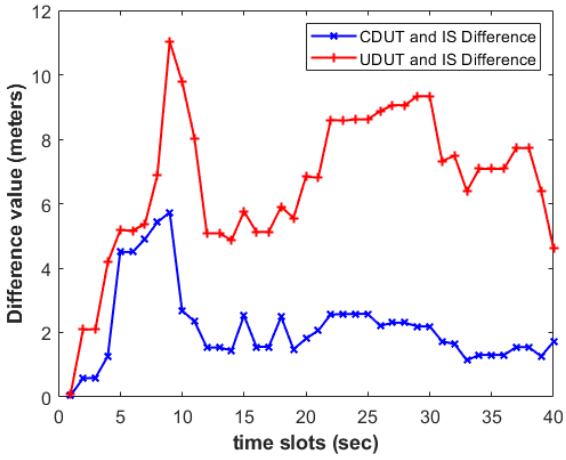

FIGURE 17. Difference between UDUT to IS and CDUT to IS.

We presented all three cases of DDPG in Figure 10 for more accurate and clear results. We also worked on flying time and communication time. Figure 11 shows flying time versus cost function. When we increase flying time, the the energy of UAV also increases that rises offload energy but we are considering normalize energy, as a result, its effect does not appear in the final normalized cost value.

Figure 12 shows the 3D trajectory of UAV of all three cases, i.e., IS, UDUT, and CDUT. CDUT algorithm minimizes the difference between IS location and UDUT location. Figure 13 shows the close view of UAV trajectory. As we can see that the proposed trajectory algorithm i.e., CDUT (training through error dynamics) follows the IS trajectory with minimum error value in contrast with UDUT. Figure 14 shows the average offloading ratio comparison of all three cases of CDUT technique. We can see that the CDUT(training through error dynamics) increases in offloading ratio and causes the total computational cost to be decreased. To demonstrate the efficacy of our approach we have also included results in which the EuDs are given random trajectory. Figure 15 shows the trajectory of the EuDs which is closely followed by the UAV. In Figure 16, it can be seen that the results are able to converge to optimal performance. Figure 17 shows the difference of the distance between trajectories. IS is taken as the reference. It is obvious that CDUT provides better accuracy.

V. CONCLUSION

In this paper, DDPG based strategy is employed for optimal computational cost, resource allocation, and 3D UAV trajectory optimization in a UAV-assisted MEC network. The presented approach considers UAV dynamics and error compensation schemes, which are often ignored in previous studies. We considered a UAV-MEC network in which each EuD offloads some portion of the task to the UAV. Based on the computational cost which is a normalized function of time delay and energy, UAV offer its services to EuDs in a way that combined optimization of trajectory and cost achieved. Also, a feedback mechanism opted in which UAV compensates its position for the next iteration with respect to the previous location.

Extensive simulations are performed and a qualitative comparison between DDPG and DQN is presented for the same conditions and parameters. The results show that the proposed DDPG based strategy is superior in compensating the error factor and cost performance as compared to DQN. In the future, we will investigate the performance of multi-UAV in a centralized and decentralized framework and observe the behavior of UAV-MEC network in the multi-dimensional objective function.

REFERENCES

[1] P. Friess and F. Ibanez, "Putting the Internet of Things forward to the next level," in *Internet of Things Applications-From Research and Innovation to Market Deployment*. Denmark: River Publishers, 2022, pp. 3–6.

[2] M. Maray and J. Shuja, "Computation offloading in mobile cloud computing and mobile edge computing: Survey, taxonomy, and open issues," *Mobile Inf. Syst.*, vol. 2022, pp. 1–17, Jun. 2022.

[3] S. M. A. Huda and S. Moh, "Survey on computation offloading in UAV-enabled mobile edge computing," *J. Netw. Comput. Appl.*, vol. 201, May 2022, Art. no. 103341.

[4] K. Sadatdiynov, L. Cui, L. Zhang, J. Z. Huang, S. Salloum, and M. S. Mahmud, "A review of optimization methods for computation offloading in edge computing networks," *Digit. Commun. Netw.*, vol. 9, no. 2, pp. 450–461, Apr. 2023.

[5] L. Zhang, W. Zhou, J. Xia, C. Gao, F. Zhu, C. Fan, and J. Ou, "DQN-based mobile edge computing for smart Internet of Vehicle," *EURASIP J. Adv. Signal Process.*, vol. 2022, no. 1, pp. 1–16, Dec. 2022.

[6] C. Yin, H. T. Nguyen, C. Kundu, Z. Kaleem, E. Garcia-Palacios, and T. Q. Duong, "Secure energy harvesting relay networks with unreliable backhaul connections," *IEEE Access*, vol. 6, pp. 12074–12084, 2018.

[7] S. S. D. Ali, H. P. Zhao, and H. Kim, "Mobile edge computing: A promising paradigm for future communication systems," in *Proc. IEEE Region Conf. (TENCON)*, Oct. 2018, pp. 1183–1187.

[8] K. Peng, V. C. M. Leung, X. Xu, L. Zheng, J. Wang, and Q. Huang, "A survey on mobile edge computing: Focusing on service adoption and provision," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–16, Oct. 2018.

[9] S. Wang, Y. Zhao, J. Xu, J. Yuan, and C.-H. Hsu, "Edge server placement in mobile edge computing," *J. Parallel Distrib. Comput.*, vol. 127, pp. 160–168, May 2019.

[10] P. Wang, C. Yao, Z. Zheng, G. Sun, and L. Song, "Joint task assignment, transmission, and computing resource allocation in multilayer mobile edge computing systems," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2872–2884, Apr. 2019.

[11] J. Zhao, Q. Li, Y. Gong, and K. Zhang, "Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7944–7956, Aug. 2019.

[12] C. Li, J. Xia, F. Liu, D. Li, L. Fan, G. K. Karagiannidis, and A. Nallanathan, "Dynamic offloading for multiuser muti-CAP MEC networks: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 70, no. 3, pp. 2922–2927, Mar. 2021.

[13] A. Shakarami, M. Ghobaei-Arani, and A. Shahidinejad, "A survey on the computation offloading approaches in mobile edge computing: A machine learning-based perspective," *Comput. Netw.*, vol. 182, Dec. 2020, Art. no. 107496.

[14] T. Alfakih, M. M. Hassan, A. Gumaei, C. Savaglio, and G. Fortino, "Task offloading and resource allocation for mobile edge computing by deep reinforcement learning based on SARSA," *IEEE Access*, vol. 8, pp. 54074–54084, 2020.

[15] J. Wang, L. Zhao, J. Liu, and N. Kato, "Smart resource allocation for mobile edge computing: A deep reinforcement learning approach," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 3, pp. 1529–1541, Jul. 2021.

[16] X. Li, L. Zhao, K. Yu, M. Aloqaily, and Y. Jararweh, "A cooperative resource allocation model for IoT applications in mobile edge computing," *Comput. Commun.*, vol. 173, pp. 183–191, May 2021.

[17] Z. Chen and X. Wang, "Decentralized computation offloading for multi-user mobile edge computing: A deep reinforcement learning approach," *EURASIP J. Wireless Commun. Netw.*, vol. 2020, no. 1, pp. 1–21, Dec. 2020.

[18] H. Shakhatreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah, and M. Guizani, "Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges," *IEEE Access*, vol. 7, pp. 48572–48634, 2019.

[19] Z. Wei, M. Zhu, N. Zhang, L. Wang, Y. Zou, Z. Meng, H. Wu, and Z. Feng, "UAV-assisted data collection for Internet of Things: A survey," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 15460–15483, Sep. 2022.

[20] G. Geraci, A. Garcia-Rodriguez, L. G. Giordano, D. López-Pérez, and E. Björnson, "Understanding UAV cellular communications: From existing networks to massive MIMO," *IEEE Access*, vol. 6, pp. 67853–67865, 2018.

[21] M. Basharat, M. Naeem, Z. Qadir, and A. Anpalagan, "Resource optimization in UAV-assisted wireless networks—A comprehensive survey," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 7, p. e4464, 2022.

[22] M. D. Nguyen, L. B. Le, and A. Girard, "UAV placement and resource allocation for intelligent reflecting surface assisted UAV-based wireless networks," *IEEE Commun. Lett.*, vol. 26, no. 5, pp. 1106–1110, May 2022.

[23] S. Wang and N. Kong, "Network resource allocation strategy based on UAV cooperative edge computing," *J. Robot.*, vol. 2022, pp. 1–9, Mar. 2022.

[24] L. Yan, C. Wang, and W. Zheng, "Secure efficiency maximization for UAV-assisted mobile edge computing networks," *Phys. Commun.*, vol. 51, Apr. 2022, Art. no. 101568.

[25] D. Zhai, H. Li, X. Tang, R. Zhang, and H. Cao, "Joint position optimization, user association, and resource allocation for load balancing in UAV-assisted wireless networks," *Digit. Commun. Netw.*, Mar. 2022, doi: 10.1016/j.dcan.2022.03.011.

[26] S. Fu, M. Zhang, M. Liu, C. Chen, and F. R. Yu, "Towards energy-efficient UAV-assisted wireless networks using an artificial intelligence approach," *IEEE Wireless Commun.*, vol. 29, no. 5, pp. 77–83, Oct. 2022.

[27] R. Adlakha and M. Zheng, "An optimization-based iterative learning control design method for UAV's trajectory tracking," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2020, pp. 1353–1359.

[28] L. V. Nguyen, M. D. Phung, and Q. P. Ha, "Iterative learning sliding mode control for UAV trajectory tracking," *Electronics*, vol. 10, no. 20, p. 2474, Oct. 2021.

[29] R. Samar, S. Ahmed, and F. Aftab, "Lateral control with improved performance for UAVs," *IFAC Proc. Volumes*, vol. 40, no. 7, pp. 37–42, 2007.

[30] G. Chowdhary, T. Wu, M. Cutler, and J. P. How, "Rapid transfer of controllers between UAVs using learning-based adaptive control," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 5409–5416.

[31] H. Nguyen, T. Quyen, C. Nguyen, A. Le, H. Tran, and M. Nguyen, "Control algorithms for UAVs: A comprehensive survey," *EAI Endorsed Trans. Ind. Netw. Intell. Syst.*, vol. 7, no. 23, May 2020, Art. no. 164586.

[32] M. Okasha, J. Kralev, and M. Islam, "Design and experimental comparison of PID, LQR and MPC stabilizing controllers for parrot mambo mini-drone," *Aerospace*, vol. 9, no. 6, p. 298, Jun. 2022.

[33] H. Zhang, L. Song, Z. Han, and H. V. Poor, "Cooperation techniques for a cellular Internet of unmanned aerial vehicles," *IEEE Wireless Commun.*, vol. 26, no. 5, pp. 167–173, Oct. 2019.

[34] S. Zhang, J. Yang, H. Zhang, and L. Song, "Dual trajectory optimization for a cooperative Internet of UAVs," *IEEE Commun. Lett.*, vol. 23, no. 6, pp. 1093–1096, Jun. 2019.

[35] A. F. Reis, G. Brante, R. Parisotto, R. D. Souza, P. H. V. Klaine, J. P. Battistella, and M. A. Imran, "Energy efficiency analysis of drone small cells positioning based on reinforcement learning," *Internet Technol. Lett.*, vol. 3, no. 5, p. e166, Sep. 2020.

[36] S. Wang, M. Chen, W. Saad, C. Yin, S. Cui, and H. V. Poor, "Reinforcement learning for minimizing age of information under realistic physical dynamics," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2020, pp. 1–6.

[37] P. Luong, F. Gagnon, L. Tran, and F. Labeau, "Deep reinforcement learning-based resource allocation in cooperative UAV-assisted wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 11, pp. 7610–7625, Nov. 2021.

[38] B. Liu, Y. Wan, F. Zhou, Q. Wu, and R. Q. Hu, "Resource allocation and trajectory design for MISO UAV-assisted MEC networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 4933–4948, May 2022.

[39] C. Liu, Y. Guo, N. Li, and X. Song, "AoI-minimal task assignment and trajectory optimization in multi-UAV-assisted IoT networks," *IEEE Internet Things J.*, vol. 9, no. 21, pp. 21777–21791, Nov. 2022.

[40] F. Xu, Z. Zhang, J. Feng, Z. Qin, and Y. Xie, "Efficient deployment of multi-UAV assisted mobile edge computing: A cost and energy perspective," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 5, p. e4453, May 2022.

[41] W. You, C. Dong, Q. Wu, Y. Qu, Y. Wu, and R. He, "Joint task scheduling, resource allocation, and UAV trajectory under clustering for FANETs," *China Commun.*, vol. 19, no. 1, pp. 104–118, Jan. 2022.

[42] K. Liu and J. Zheng, "UAV trajectory optimization for time-constrained data collection in UAV-enabled environmental monitoring systems," *IEEE Internet Things J.*, vol. 9, no. 23, pp. 24300–24314, Dec. 2022.

[43] K. K. Nguyen, T. Q. Duong, T. Do-Duy, H. Claussen, and L. Hanzo, "3D UAV trajectory and data collection optimisation via deep reinforcement learning," *IEEE Trans. Commun.*, vol. 70, no. 4, pp. 2358–2371, Apr. 2022.

[44] S. Javed, A. Hassan, R. Ahmad, W. Ahmed, M. M. Alam, and J. J. P. C. Rodrigues, "UAV trajectory planning for disaster scenarios," *Veh. Commun.*, vol. 39, Feb. 2023, Art. no. 100568.

[45] M. Waheed, R. Ahmad, W. Ahmed, M. M. Alam, and M. Magarini, "On coverage of critical nodes in UAV-assisted emergency networks," *Sensors*, vol. 23, no. 3, p. 1586, Feb. 2023.

[46] J. Huang, S. Xu, J. Zhang, and Y. Wu, "Resource allocation and 3D deployment of UAVs-assisted MEC network with air-ground cooperation," *Sensors*, vol. 22, no. 7, p. 2590, Mar. 2022.

[47] A. M. Seid, G. O. Boateng, B. Mareri, G. Sun, and W. Jiang, "Multi-agent DRL for task offloading and resource allocation in multi-UAV enabled IoT edge network," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 4, pp. 4531–4547, Dec. 2021.

[48] A. M. Seid, G. O. Boateng, S. Anokye, T. Kwantwi, G. Sun, and G. Liu, "Collaborative computation offloading and resource allocation in multi-UAV-assisted IoT networks: A deep reinforcement learning approach," *IEEE Internet Things J.*, vol. 8, no. 15, pp. 12203–12218, Aug. 2021.

[49] W. Fang, S. Ding, Y. Li, W. Zhou, and N. Xiong, "OKRA: Optimal task and resource allocation for energy minimization in mobile edge computing systems," *Wireless Netw.*, vol. 25, no. 5, pp. 2851–2867, Jul. 2019.

[50] J. Xiong, H. Guo, and J. Liu, "Task offloading in UAV-aided edge computing: Bit allocation and trajectory optimization," *IEEE Commun. Lett.*, vol. 23, no. 3, pp. 538–541, Mar. 2019.

[51] U. Saleem, Y. Liu, S. Jangsher, and Y. Li, "Performance guaranteed partial offloading for mobile edge computing," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[52] Q. Hu, Y. Cai, G. Yu, Z. Qin, M. Zhao, and G. Y. Li, "Joint offloading and trajectory design for UAV-enabled mobile edge computing systems," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1879–1892, Apr. 2019.

[53] N. T. Ti and L. B. Le, "Joint resource allocation, computation offloading, and path planning for UAV based hierarchical fog-cloud mobile systems," in *Proc. IEEE 7th Int. Conf. Commun. Electron. (ICCE)*, Jul. 2018, pp. 373–378.

[54] H. Zhang, Z. Han, and H. V. Poor, "Trajectory optimization for UAV-to-device underlaid cellular networks by mean-field-type control," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2020, pp. 1–6.

[55] M. Laskin, K. Lee, A. Stooke, L. Pinto, P. Abbeel, and A. Srinivas, "Reinforcement learning with augmented data," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 19884–19895.

[56] J. Zeng, D. Ding, K. Kang, H. Xie, and Q. Yin, "Adaptive DRL-based virtual machine consolidation in energy-efficient cloud data center," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 11, pp. 2991–3002, Nov. 2022.

[57] S. Ayas and M. S. Ayas, "A novel bearing fault diagnosis method using deep residual learning network," *Multimedia Tools Appl.*, vol. 81, no. 16, pp. 22407–22423, Jul. 2022.

[58] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1995–2003.

[59] S. Yin, S. Zhao, Y. Zhao, and F. R. Yu, "Intelligent trajectory design in UAV-aided communications with reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8227–8231, Aug. 2019.

[60] W. Luo, Q. Tang, C. Fu, and P. Eberhard, "Deep-SARSA based multi-UAV path planning and obstacle avoidance in a dynamic environment," in *Advances in Swarm Intelligence*, Y. Tan, Y. Shi, and Q. Tang, Eds. Cham, Switzerland: Springer, 2018, pp. 102–111.

[61] S. Mohamed and R. Ejbali, "Deep SARSA-based reinforcement learning approach for anomaly network intrusion detection system," *Int. J. Inf. Secur.*, vol. 22, no. 1, pp. 235–247, Feb. 2023.

[62] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. AAAI Conf. Artif. Intell.*, 2016, vol. 30, no. 1, pp. 2094–2100.

[63] J. Haseeb, S. U. R. Malik, M. Mansoori, and I. Welch, "Probabilistic modelling of deception-based security framework using Markov decision process," *Comput. Secur.*, vol. 115, Apr. 2022, Art. no. 102599. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404821004223

[64] Y. Liu, H. Liang, Y. Xiao, H. Zhang, J. Zhang, L. Zhang, and L. Wang, "Logistics-involved service composition in a dynamic cloud manufacturing environment: A DDPG-based approach," *Robot. Comput.-Integr. Manuf.*, vol. 76, Aug. 2022, Art. no. 102323.

[65] M. Hossny, J. Iskander, M. Attia, K. Saleh, and A. Abobakr, "Refined continuous control of DDPG actors via parametrised activation," *AI*, vol. 2, no. 4, pp. 464–476, Sep. 2021.

[66] S. Lee, S. Jin, S. Hwang, and I. Lee, "Learning optimal trajectory generation for low-cost redundant manipulator using deep deterministic policy gradient (DDPG)," *J. Korea Robot. Soc.*, vol. 17, no. 1, pp. 58–67, Mar. 2022.

[67] Z. Zhang, Q. Zhang, J. Miao, F. R. Yu, F. Fu, J. Du, and T. Wu, "Energy-efficient secure video streaming in UAV-enabled wireless networks: A safe-DQN approach," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 4, pp. 1892–1905, Dec. 2021.

[68] F. Wu, H. Zhang, J. Wu, L. Song, Z. Han, and H. V. Poor, "AoI minimization for UAV-to-device underlay communication by multi-agent deep reinforcement learning," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2020, pp. 1–6.

[69] M. Coldrey, J.-E. Berg, L. Manholm, C. Larsson, and J. Hansryd, "Non-line-of-sight small cell backhauling using microwave technology," *IEEE Commun. Mag.*, vol. 51, no. 9, pp. 78–84, Sep. 2013.
[70] J. Nie and S. Haykin, "A Q-learning-based dynamic channel assignment technique for mobile communication systems," *IEEE Trans. Veh. Technol.*, vol. 48, no. 5, pp. 1676–1687, Sep. 1999.
[71] S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2049–2063, Mar. 2018.
[72] W. A. Reid and I. M. Albayati, "Design of an unmanned aircraft system for high-altitude 1 kW fuel cell power system," *Aerosp. Syst.*, vol. 4, no. 4, pp. 353–363, Dec. 2021.
[73] F. Hsieh, F. Jardel, E. Visotsky, F. Vook, A. Ghosh, and B. Picha, "UAV-based multi-cell HAPS communication: System design and performance evaluation," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2020, pp. 1–6.

TAYYABA KHURSHID received the M.E. degree in telecommunication engineering from the NED University of Engineering and Technology, Karachi, Pakistan, in 2017. She is currently pursuing the Ph.D. degree in electrical engineering with the Pakistan Institute of Engineering and Applied Science (PIEAS), Islamabad, Pakistan. Since 2020, she has been a Lecturer with the Department of Telecommunication Engineering, Dawood University of Engineering and Technology, Karachi. Her research interests include resource allocation in wireless networks, handover in uplink-downlink decoupling, heterogeneous cellular networks, spectrum sensing, and cognitive radio networks.

WAQAS AHMED received the M.S. degree in systems engineering from the Pakistan Institute of Engineering and Applied Sciences, in 2005, and the Ph.D. degree in electrical engineering from Victoria University, Melbourne, Australia, in 2012. Since 2007, he has been a Professor with the Department of Electrical Engineering, Pakistan Institute of Engineering and Applied Sciences. He has published and served as a reviewer for IEEE journals and conferences. His research interests include cognitive radios, cooperative communication, and physical layer aspects of wireless communication and networks. He also serves on the TPC for leading conferences in the communication and networking field, including IEEE VTC, IEEE ICC, and IEEE WCNC. He was a recipient of the IEEE Exemplary Reviewer Award, in 2010.

MUHAMMAD REHAN (Member, IEEE) received the M.Sc. degree in electronics from Quaid-e-Azam University (QAU), Islamabad, the M.S. degree in systems engineering from the Pakistan Institute of Engineering and Applied Sciences (PIEAS), Islamabad, and the Ph.D. degree (Hons.) from the Department of Cogno-Mechatronics Engineering, Pusan National University, Busan, Republic of Korea, in 2012. He is currently a Full Professor with the Department of Electrical Engineering, PIEAS. His research interests include robust control, nonlinear, adaptive control, anti-windup design, modeling and control of bio-systems, control of multi-agents, and distributed optimization over a networks. He received the Research Productivity Award by the Pakistan Council of Science and Technology, for the years 2011–2012, 2015–2016, and 2016–2017. He has been selected as a Young Associate in the discipline of engineering by the Pakistan Academy of Sciences in a nationwide competition. He has been selected for the Best Young Research Scholar Award (Pure Engineering) and received the Best Paper Award in the 5th Outstanding Research Award by HEC in a nationwide competition. Recently, he has been selected in the list of top 2% scientists in the world for the year 2019, prepared by Stanford University. He is an Associate Editor of the *International Journal of Control, Automation and Systems* and *Results in Control and Optimization*.

RIZWAN AHMAD received the M.Sc. degree in communication engineering and media technology from the University of Stuttgart, Stuttgart, Germany, in 2004, and the Ph.D. degree in electrical engineering from Victoria University, Melbourne, Australia, in 2010. From 2010 to 2012, he was a Postdoctoral Research Fellow with Qatar University on a QNRF Grant. He is currently a Professor and the HoD with the School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Pakistan. He has published and served as a reviewer for IEEE journals and conferences. His research interests include public safety networks, medium access control protocols, spectrum and energy efficiency, energy harvesting, and performance analysis for wireless communication and networks. He was a recipient of the prestigious International Postgraduate Research Scholarship from the Australian Government.

MUHAMMAD MAHTAB ALAM (Senior Member, IEEE) received the M.Sc. degree in electrical engineering from Aalborg University, Denmark, in 2007, and the Ph.D. degree in signal processing and telecommunication from the University of Rennes 1 (INRIA Research Center), France, in 2013. He did his postdoctoral research (2014–2016) with the Qatar Mobility Innovation Center, Qatar. In 2016, he joined as the European Research Area Chair and as an Associate Professor with the Thomas Johann Seebeck Department of Electronics, Tallinn University of Technology, where he was elected as a Professor, in 2018, and a tenured Full Professor, in 2021. Since 2019, he has been the Communication Systems Research Group Leader. He has over 15 years of combined academic and industrial multinational experiences while working in Denmark, Belgium, France, Qatar, and Estonia. He has several leading roles as PI in multimillion Euros international projects funded by European Commission (Horizon Europe LATEST-5GS, 5G-TIMBER, H2020 5G-ROUTES, NATOSPS (G5482), Estonian Research Council (PRG424), and Telia Industrial Grant. He is the author or coauthor of more than 100 research publications. He is actively supervising a number of Ph.D. and postdoctoral researchers. He is also a contributor to two standardization bodies (ETSI SmartBAN and IEEE-GeeenICT-EECH), including "Rapporteur" of work item: DTR/SmartBAN-0014. His research interests include wireless communications connectivity, mobile positioning, and 5G/6G services and applications.

AYMAN RADWAN (Senior Member, IEEE) received the M.A.Sc. degree in systems and computer engineering, with an emphasis on DSP from Carleton University, Ottawa, ON, Canada, in 2003, and the Ph.D. degree in electrical and computer engineering, focusing on wireless networking from Queen University, Kingston, ON, Canada, in 2009. He is currently an Assistant Professor with Universidade de Aveiro and a Senior Researcher with Institute de Telecomunicações, Aveiro, Portugal. He is intensively active in EU projects. He has acted as the coordinator of multiple EU joint research projects, with international partners. He is also coordinating the EU Project CELTIC-NEXT SAFE-HOME, with an emphasis on eHealth, and energy-efficient fog-cloud networking. He was involved in multiple successful proposals, raising more than U.S. $2 million in funding for his institute. He has more than 150 published highly cited peer-reviewed articles. His research interests include network architectures (specifically 5G and beyond), fog-cloud networking, the IoT, and eHealth. He is acting as the Secretary of the IEEE eHealth TC.