

Multi-Agent Deep Reinforcement Learning for Wireless-Powered UAV Networks

Omar Sami Oubbati, Abderrahmane Lakas, Mohsen Guizani

▶ **To cite this version:**

Omar Sami Oubbati, Abderrahmane Lakas, Mohsen Guizani. Multi-Agent Deep Reinforcement Learning for Wireless-Powered UAV Networks. IEEE Internet of Things Journal, 2022, 9 (17), pp.16044-16059. 10.1109/JIOT.2022.3150616 . hal-03581840

HAL Id: hal-03581840
https://hal.science/hal-03581840

Submitted on 20 Feb 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Multi-Agent Deep Reinforcement Learning for Wireless-Powered UAV Networks

Omar Sami Oubbati, *Member, IEEE*, Abderrahmane Lakas, *Senior Member, IEEE*, and Mohsen Guizani, *Fellow, IEEE*

***Abstract*—Unmanned Aerial Vehicles (UAVs) have attracted much attention lately and are being used in a multitude of applications. But the duration of being in the sky remains to be an issue due to their energy limitation. In particular, this represents a major challenge when UAVs are used as base stations (BSs) to complement the wireless network. Therefore, as UAVs execute their missions in the sky, it becomes beneficial to wirelessly harvest energy from external and adjustable flying energy sources (FESs) to power their onboard batteries and avoid disrupting their trajectories. For this purpose, wireless power transfer (WPT) is seen as a promising charging technology to keep UAVs in flight and allow them to complete their missions. In this work, we leverage a multi-agent deep reinforcement learning (MADRL) method to optimize the task of energy transfer between FESs and UAVs. The optimization is performed by carrying out three essential tasks: (i) maximizing the sum-energy received by all UAVs based on FESs using WPT, (ii) optimizing the energy loading process of FESs from a ground BS, and (iii) computing the most energy-efficient trajectories of the FESs while carrying out their charging duties. Furthermore, to ensure high-level reliability of energy transmission, we use directional energy transfer for charging both FESs and UAVs by using laser beams and energy beam-forming technologies, respectively. In this study, the simulation results show that the proposed MADRL method has efficiently optimized the trajectories and energy consumption of FESs, which translates into a significant energy transfer gain compared to the baseline strategies.**

***Index Terms*—UAV; Wireless Power Transfer (WPT), Energy Harvesting; Deep Reinforcement Learning; Energy Efficiency.**

I. Introduction

The development of Unmanned Aerial Vehicles (UAVs) have led to the emergence of an extensive range of UAV-enabled applications and services spanning from parcel delivery and public safety to monitoring and disaster management [1]. Deploying UAVs as flying base stations (BSs) could take several forms, such as aerial hot-spots or small cell networks, adjusting their locations dynamically and extending the coverage and capacity of ground cellular networks [2], [3]. However, the operating time of UAVs is severely limited by their short-lived built-in only energy sources (*e.g.,* batteries) [4]. Thus, some existing works in the literature have adopted various energy-efficient strategies to prolong the flying time of UAV networks by improving specific parameters such as trajectories, end-to-end communications, and transmission power, or by using replacement strategies for the deployed UAVs [5]–[7]. But, unfortunately, these improvements remain limited and do not sufficiently extend the UAV network's lifetime [8]. Moreover, this constraint leads the onboard batteries to be replaced or recharged periodically, resulting in a high cost in terms of mission delay and incurred interruptions, thus, affecting the entire UAV network performance.

To guarantee uninterrupted UAV missions and meet the challenge of endless energy supply, wireless power transfer (WPT) technology is considered a leading solution to providing flexible and cost-effective energy supply to UAVs [9]. Indeed, the far-field WPT technology based on RF signals is considered as a promising approach for powering UAVs [10]. Therefore, WPT technology is expected to have abundant applications in the future generation of mobile networks, such as 5G, Beyond 5G, and 6G (*see* [11], [12] and the references therein). In general, a typical WPT system consists of a set of energy receivers (ERs) (*e.g.,* low-powered Internet of Things (IoT) devices, sensors, or even UAVs) harvesting energy that is wirelessly emitted from static energy transmitters (ETs) that are dispatched at fixed locations. Nevertheless, the performance of WPT systems is seriously degraded when there are long distances between ETs and ERs, which is due to the problem of limited power transmission resulting in severe propagation loss of radio frequency (RF) signals. In the case of ER-equipped UAVs, two options could be envisaged to address this issue. Firstly, an important number of ETs could be massively deployed [13] on the ground. However, this option would significantly increase the cost and UAVs have to fly near ETs, forcing UAVs to divert from their mission and fly at low altitudes during the energy loading process from ETs using WPT. Unfortunately, this option is not always possible, and especially when ETs are deployed in an environment full of obstacles (*e.g.,* forests or mountainous areas), where the energy harvesting using WPT technology could suffer from the non-line-of-sight (NLoS) problem, which significantly decreases the efficiency of energy transfer.

This research is motivated by the need to find a potential solution to power UAVs while accomplishing their missions with the objective to overcome three main challenges. First, UAVs operating in remote regions need to recharge or replace their batteries regularly, which requires a massive deployment of landing/charging stations (CSs) to avoid irreversible UAV failure and the need of human intervention. This solution will be costly in installing and maintaining CSs and could lead to numerous drawbacks, namely when the deployment area is constrained. Second, even if some existing solutions have

O.S. Oubbati is with LIGM, University Gustave Eiffel, Marne-la-Vallée, France. E-mail: omar-sami.oubbati@univ-eiffel.fr

A. Lakas is with College of Information Technology, United Arab Emirates University, United Arab Emirates. Email: alakas@uaeu.ac.ae

M. Guizani is with the Machine Learning Department, Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI), United Arab Emirates. E-mail: mguizani@ieee.org

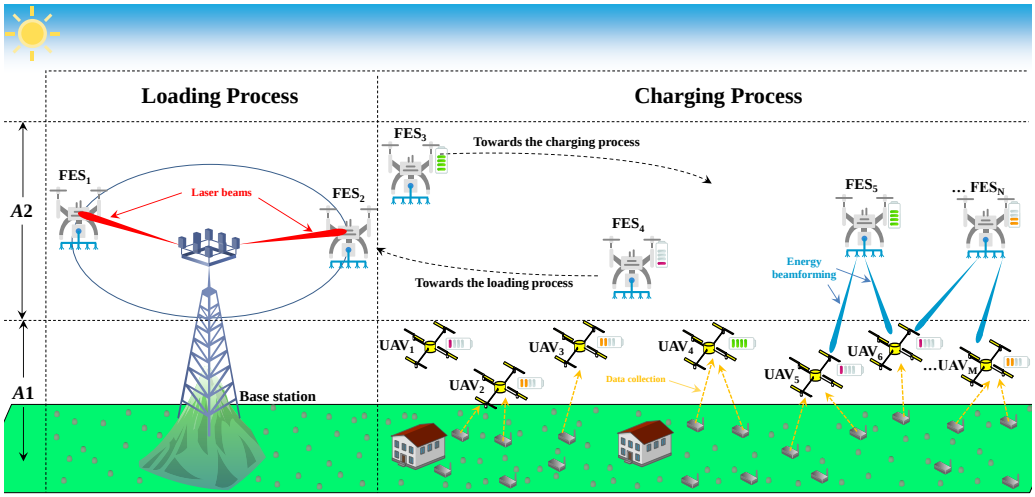


Fig. 1: Motivating scenario.

attempted to optimize the density and deployment of CSs, they do not prevent the interruption of missions performed by UAVs that need to go back and forth to recharge/replace their batteries. This significantly lengthens the missions' completion time and affects the proper functioning of the network, especially when UAVs are deployed to serve ground users. Finally, conventional far-field WPT systems suffer from many problems, including limited transfer distance, energy attenuation due to high path loss, unpredictable reliability, and other issues [14]–[16]. This results in reducing the WPT efficiency and thus minimizing the amount of energy harvested by UAVs. A possible solution should address all the issues mentioned previously and improve the energy use and the WPT performance. Our proposed solution is inspired by air-to-air refueling of military jets using aerial tankers [17]. In this solution, we propose to deploy a set of intelligent flying energy sources (FESs) operating autonomously with the sole objective of recharging UAVs efficiently (*see* Fig. 1). FESs are themselves UAVs with onboard WPT ETs, and therefore they are supposed to have a higher energy capacity allowing them to transfer energy to regular ER-equipped UAVs. FESs being themselves UAVs, are recharged from ground CSs. The FESs appropriately maintain LoS of RF links and reduce distances to UAVs for proper charging using energy beamforming. However, the deployment of FESs faces serious challenges. For example, since UAVs with a low energy level can appear anywhere and at any time, FESs are required to continuously hover and cater to the UAVs' energy need. As a result, FESs share a common flight space (*see* A2 in Fig. 1), leading to potential collisions with all flying vehicles and other surrounding obstacles (*e.g.,* high-rise buildings or communication towers). The recharging process along with constant mobility and communication to coordinate with various entities leads inevitably to excessive energy consumption. Moreover, UAV trajectories are solely decided by the nature of their missions – e.g., collecting data from ground devices and sensors (*see* A1 in Fig. 1), FESs find themselves adapting their own trajectories to minimize the gap with UAVs to execute the energy transfer efficiently. To address the issues of classical far-field WPT systems, we adopt energy beam-forming technology, which presents itself as an efficient solution that maximizes the received signal strength. Therefore, FESs (*c.f.,* Fig. 1) are equipped with an antenna array to simultaneously establish multiple energy beams towards UAVs depending on their locations, and therefore enhance the energy transfer efficiency.

It is worth noting that it becomes a complex task to jointly consider the challenges mentioned above when trying to solve a single problem optimally using existing optimization techniques. That is, we rely on the use of Deep Reinforcement Learning (DRL) methods applied to multi-agent environments, which have demonstrated their efficiency to process large state space and time-varying environments. Furthermore, DRL methods can provide near-optimal performance on several learning tasks with little or no domain knowledge. The main contributions of this article are summarized as follows:

- Designing a wirelessly-powered UAV network architecture based on autonomous mobile FESs that provide energy supply to moving UAVs without disrupting their trajectories and/or missions.
- Optimizing the trajectories of FESs, avoiding collisions between themselves, and ensuring an acceptable level of fairness when recharging UAVs.
- Optimizing FESs energy loading process from the ground CS while taking into account the energy requirements of the FESs.
- Providing an analytical and numerical basis for the validation of the proposed approach and the analysis of its effectiveness.

The rest of this paper is organized as follows. Section II discusses the related literature papers that we believe are relevant to our work. Section III introduces a novel wireless powered UAV network architecture and formulates the problem statement. Section IV provides a set of DRL preliminaries and describes the elements of our multi-agent DRL-based approach. In Section V, we present a performance evaluation and analysis of the proposed solution with a description of the simulation results. Section VI concludes this study.

II. Related Work

There exists incredible array of energy limited IoT devices, including UAVs [18]. As a promising solution, WPT has shown its advantages in supplying energy consumption in many UAV-assisted applications. Indeed, according to [19], the WPT market is set to exceed $12 billion by 2022 and will grow to reach $25 Billion by 2025. Similarly, based on a recent study appeared in [20], the UAV market has also seen a meteoric growth that would reach $43 billion by 2025 (*c.f.,* Fig. 2).

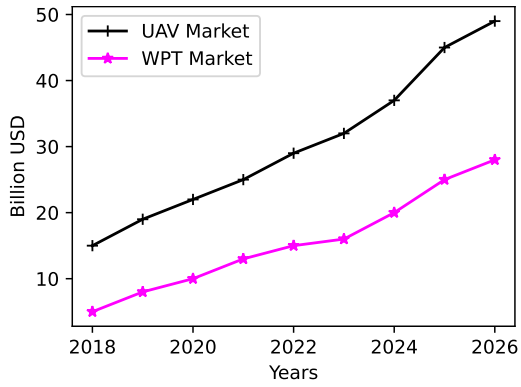


Fig. 2: Statistic of UAV and WPT market growth (Data Source: Drone Industry Insight Report 2020-2025 [21]).

From this statistic, it is only natural to expect UAVs to use WPT technology in an integrated manner and to power ERs. To stay within the scope of this work, we focus our interest here around three major research areas related to UAV-enabled WPT systems: (i) Terrestrial CS, (ii) UAV trajectory optimization, and (iii) Reinforcement Learning (RL) based techniques for optimizing UAV-enabled WPT systems.

A. Terrestrial CS-based WPT

CSs are quite necessary to allow UAVs to perform a long-term mission. For instance, in [22], the authors deployed wireless powered UAV-BSs for data transmissions to ground users (GUs). To maximize the downlink sum rate of the system, the placement of UAV-BSs, the resource allocations of energy, and the time duration for TDMA and FDMA are jointly optimized. The work of [23] maximized the coverage of UAVs, while optimizing UAV 3D deployment and scheduling energy recharging actions of UAVs. In [24], a UAV-based data collection scheme is designed, which is energetically supported by a mobile CS on the ground. Liu *et al.* [25] have deployed a wireless UAV network, where each UAV has to be serviced before deployment. In another study [26], the authors focused on minimizing the number of CSs and optimizing their deployment where UAVs can recharge their batteries and then fly again. Also, the influence of CSs on network performance is studied in [27]. This study concluded that achieving an optimized coverage can be reached by reducing the charging time and deploying lower dense CSs.

Existing terrestrial CS-based approaches have demonstrated their efficiency in improving the endurance of the deployed UAVs. However, UAVs should interrupt their missions and flight back onto a CS, which prevents them from continuing their tasks during a longer charging time. This constraint could constitute an inflexibility issue, especially in emergencies or time-sensitive applications.

B. UAV-enabled Trajectory-aware WPT

Much research has been conducted to optimize UAV trajectories to optimize the energy transfer process. For instance, in [28], the speed of UAVs is optimized to maximize the uplink throughput of a UAV-enabled WPT system. Hu *et al.* [29] maximized the harvested energy among GUs, while considering the maximum UAV speed constraints. In [30], the UAV data collectors' energy consumption is reduced based on the energy harvested from the surroundings. The work in [31] enhanced the efficiency of a UAV-assisted wireless powered communication network (WPCN) by considering the UAV mission time and energy consumption of UAVs. In [32], two UAV deployment's scenarios are studied: (i) a single UAV to perform both the energy transfer and information harvesting towards and from GUs and (ii) Two different UAVs make the two tasks separately. The two scenarios aimed to maximize the minimum throughput of GUs. Wu *et al.* [33] studied the UAV's trajectory optimization problem intending to design a trajectory that should maximize the energy utilization efficiency and thus prolong the lifetime of sensor networks. Another UAV-assisted wireless powered sensor network is proposed in [34]. The authors used a heuristic algorithm to optimize the trajectory of the UAV while considering some channel parameters.

However, various constraints are neglected in UAV-enabled trajectory-aware WPT approaches. For example, some techniques are based only on a single UAV to perform energy transfer. In contrast, the other techniques are based on multiple UAVs without considering some issues, such as collisions, energy transfer fairness, and completion time. In addition, the UAV energy consumption is often neglected, which makes the proposed approaches not realistic.

C. RL-enabled WPT

Recently, we have witnessed a resurgence of interest in machine learning (ML) techniques, and more particularly, RL methods in the context of UAV-enabled WPT systems. In [35], a Q-learning method is applied to address the fairness in terms of energy transfer in a UAV-enabled WPT system. The authors of [39] deployed a stationary FES to charge UAV-ERs. This architecture aims to optimize the location of UAV-ET to maximize the total harvested energy by UAV-ERs, which are flying in a linear trajectory in a square area. Nevertheless, the authors assume that UAV-ET and UAV-ERs fly at the same altitude, which can increase the probability of collision between each other. To address this issue, Hosseini *et al.* [36] proposed to maximize the long-term flying time of UAV-ERs by optimizing the energy transfer process. Since there are many UAV-ERs, the authors propose to optimize the trajectory of a single UAV-ET based on the Q-learning method. Similarly, in [37], the authors deployed an aerial charging host to maintain UAV-ERs in flight, which is formulated as a Q-learning problem

TABLE I: Features comparison of the related WPT-enabled schemes.

	Terrestrial CS-based		UAV Trajectory-aware		RL-enabled				
Features	Ref. [22]	Ref. [24]	Ref. [28]	Ref. [29]	Ref. [35]	Ref. [36]	Ref. [37]	Ref. [38]	Proposed method
Basic ideology	Resource allocation in wireless powered UAV-assisted cellular network.	UAV-enabled data collection with the help of mobile CS.	Throughput maximization through UAV-enabled energy harvesting system.	UAV trajectory optimization for maximizing GUs harvested energy.	Q-learning based trajectory optimization of UAV-enabled WPT system.	Q-learning based energy transfer and flying time optimization.	Q-learning based optimal charging sequence of UAV-ERs.	DRL-based resource management of UAV-enabled WPT system.	DRL-based wireless powered UAV network.
Energy Transfer Technology	RF-signal	RF-signal	RF-signal	RF-signal	RF-signal	Energy beamforming	RF-signal	RF-signal	Laser/Energy beamforming
Optimization technique	Successive convex	DRL	Successive convex	Lagrange dual	Q-learning	Q-learning	Q-learning	DRL	DRL
Type of ET(s)	Terrestrial CS	Terrestrial CS	Flying CS	Flying CS	Flying CS	Flying CS	Flying CS	Flying CS	Flying CS
Type of ER(s)	UAV	UAV	GU	GU	UAV	UAV	UAV	IoT	UAV
Density of ET(s)	A single CS	A single CS	A single CS	A single CS	A single CS	A single CS	A single CS	A single CS	Multiple CSs
Density of ER(s)	A single UAV	A single UAV	Multiple GUs	Multiple GUs	Two UAVs	Multiple UAVs	Multiple UAVs	Multiple IoT	Multiple UAVs
Mobility of ET(s)	Static	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic	Dynamic
Mobility of ER(s)	Dynamic	Dynamic	Static	Static	Static	Static	Static	Static	Dynamic
Major advantage	Resource is equally shared among GUs.	Providing sufficient energy for cruising UAV	Uplink throughput is optimized and fairness issue is addressed	Minimal received energy is maximized among GUs	Less complexity	Level of received wireless power is enhanced.	The stability of the whole system is maintained	Data packet loss is minimized	Maintain the integrality of UAVs in flight
Major Limitation	Placement of a UAV-BS	Unexpected events are not considered	Energy consumption of UAV is neglected	Significant complexity is induced	less scalable	Assumptions are not realistic	Mobility of UAV-ERs is not considered	Both dynamic battery capacity and queue size are not considered	More complex

to find the optimal charging sequence of UAV-ERs. In [38], the authors investigated the resource allocation problem in UAV-enabled WPT and data collection for minimizing both data packet loss and energy consumption of IoT nodes. For this purpose, a DRL-based resource management method is adopted, which allows the UAV to optimally determine the data collection sequences, the power transmission, and the associated modulation scheme of IoT nodes. The authors of [40] deployed a single UAV to perform, at the same time, data collection and wireless power transfer towards ground nodes. Both tasks are optimized based on a DRL method. In the same way, a multi-objective optimization is proposed in [41] where a UAV is deployed to collect data from a target device and charge other covered devices. Using an adequate DRL variant, this approach jointly optimizes three objectives during a given mission period: (i) maximization of sum data rate, (ii) maximization of total harvested energy, and (iii) minimization of UAV's energy consumption.

As for drawbacks, all these research attempts focus only on small and straightforward scenarios, and most with just a single UAV that plays the role of FES. However, for larger scenarios (*i.e.,* scenarios with hundreds or thousands of UAVs acting as ERs), multiple FESs are required to reasonably satisfy the UAVs' energy requirements, where DRL methods are more appropriate to support complex and broad-scale environments.

In this context, this paper concentrates on designing a wirelessly powered UAV network architecture by flexibly deploying multiple FESs to maintain all UAVs in flight, and thus stabilize the whole system for a long period of time. Furthermore, to increase the efficiency of the energy transfer, we exploit energy beamforming technology to transfer power to UAVs directionally, while finding an optimal policy to maximize the benefit of the system. As far as we know, it is the first attempt to optimize the trajectories of FESs based on DRL methods to maximize the total harvested energy by UAVs, while considering their mobility and the different constraints that could increase the stability of the system. To recapitulate, Table I provides a brief comparative study based on crucial parameters between our proposed scheme and the most relevant schemes previously described.

III. System Model

As illustrated in Fig. 1, we assume that the architecture of our wireless powered UAV network consists of a set $\mathcal{M} \triangleq \{m = 1, 2, \ldots, M\}$ of UAVs, which are deployed in an agricultural area to perform data collection from ground sensors. To prevent UAVs from draining their batteries and falling to the ground, another set $\mathcal{N} \triangleq \{n = 1, 2, \ldots, N\}$ of FESs with highly efficient batteries are deployed to perform the charging process of UAVs, where $1 \leq |\mathcal{N}| < |\mathcal{M}|$. To avoid collisions between the two sets, FESs and UAVs are supposed to move freely in a 3D space at different altitude intervals (*i.e.,* flying spaces), namely $A2$ and $A1$, respectively. To wirelessly supply FESs with energy, a terrestrial BS without a dedicated landing dock is implemented on a hill at the edge of a target square area of width w. The BS has a height of h_{BS} from the ground, *i.e.,* its coordinates is denoted as $l_{BS} = [0, 0, h_{BS}]$. The duration of the UAV data collection mission is denoted as $t \in [0, Fly]$ in which our architecture is tested. To streamline the test, the mission duration is discretized into T time-slots, where $\psi = \frac{Fly}{T}$ is the length of each time-slot $t \in \mathcal{T} \triangleq \{t = 1, 2, \ldots, T\}$. T is supposed to be sufficiently large such that FESs tend to be pretty much still at each time-slot. The locations of each UAV$_m$, where $m \in \mathcal{M}$, is denoted as $l_m[t] = [x_m[t], y_m[t], h_m[t]]^T$ at each time-slot t. We assume that each UAV$_m$ is moving strictly inside the target area. All FESs are continuously changing their locations, looking for UAVs with low energy levels to recharge them in a timely manner. The instantaneous locations of each FES$_n$, at each time-slot, is denoted by $l_n[t] = [x_n[t], y_n[t], h_n[t]]^T$. It should be stressed that the altitudes of both FESs and UAVs are dynamic according to the positions of sensors installed at fixed positions on non-flat terrains, which are denoted as $h_m[t]$ and $h_n[t]$, respectively. The distance between each UAV$_m$ and FES$_n$ is calculated as follows:

$$d_m^n[t] = \sqrt{\|l_m[t] - l_n[t]\|^2} \tag{1}$$

It is assumed that the movements of FESs are controlled by the BS to which FESs are backhaul-connected through FES-to-

BS links. Indeed, FESs should adapt their trajectories according to those of UAVs such that they can shorten the distance and improve the line of sight (LoS) of RF links, meet the energy needs of UAVs, and maintain them in flight during the whole mission or even for long periods. Due to the restricted number of FESs and the scalable nature of the UAV network, FESs need to continuously fly around to charge UAVs with drained batteries. It is worth noting that we prioritize charging specific UAVs over others, according to their energy levels. Moreover, despite the particular deployment of UAVs in this work, the proposed trajectory optimization of FESs will hold applicable for any multi-UAV-assisted WPT applications. To summarize, our approach is executed based on two processes: (i) the loading process where FESs get their energy supplies from the BS and (ii) the charging process where FESs revolve around UAVs to recharge them with energy. For more clarity, the list of major notations used in this paper is given in Table II.

TABLE II: List of notations.

Label	Explanation
$\mathcal{T}, T, t$	Set, number, and index of time-slots
$\mathcal{N}, N, n$	Set, number, and index of FESs
$\mathcal{M}, M, m$	Set, number, and index of UAVs
$d_n^m[t]$	Distance between nodes n and m
$E_m^n[t]$	Energy harvested at node m from node n
$l_n[t], E_n[t]$	Coordinates, and residual energy of node n
$\Upsilon_n, EL_n[t]$	Transmission power and energy level of node n
E_n^{max}, V_n^{max}	Maximum energy and speed capacity of node n
$\omega_n[t], h_n[t], d_n[t]$	Flying direction, altitude, and distance of node n
$P_n^m[t], G_n^m[t]$	Signal strength, channel gain between n and m
$Ac_n[t], Nb_n[t]$	Active and fully charged status of node n
$UC[t], \sigma^t$	Fully charged density, energy balancing of nodes
$PR[t], F[t]$	Energy transfer priorities, and fairness
$PR_n^m[t], SER_n^m[t]$	Energy transfer priority, charging service status
$LD_n[t], PR_n[t]$	Service load, and energy transfer priority of n
$r_t^n, \rho_n[t]$	Reward and penalty of node n
o_t^n, a_t^n	Observation and action of node n
s_t, a_t	States and actions of the set $\mathcal{N}$
$\pi^n(.), Q^n(.)$	Actor and critic networks
$\pi^{n'}(.), Q^{n'}(.)$	Target actor and critic networks
η^{Q^n}, η^{π^n}	Parameters of critic and actor networks
$\eta^{Q^{n'}}, \eta^{\pi^{n'}}$	Parameters of target critic and actor networks
$\mathcal{B}_n, \Delta, \delta$	Buffer, size and index of Mini-batch of node n
ε, ν	Action noise and discount factor

A. Channel Modeling

To illustrate the mechanism of energy harvesting and the channel modeling of our architecture, we consider two channels: (i) BS-to-FES channel and (ii) FES-to-UAV channel.

In the literature, laser power has become a promising solution to provide a convenient energy supply to FESs. Indeed, as illustrated in Fig. 3, we assume that the BS is composed of N laser beam directors (LBDs). To power the FESs, LBDs transmit concentrated laser beams towards their receiver telescopes based on a wireless channel dominated by line-of-sight (LoS) links. Let us take, as an example, an LBD_n transmitting energy towards FES_n with a fixed power $\Upsilon_{LBD_n} > 0$. The distance between the two devices can be expressed as $d_n^{LBD_n}[t] = \sqrt{Z_n[t]^2 + J_n[t]^2}$, where $Z_n[t]$ and

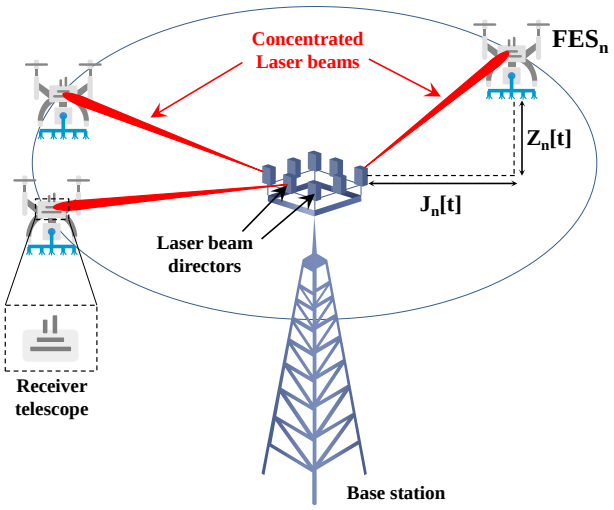


Fig. 3: Channel modeling of BS-enabled loading process.

$J_n[t]$ are the flight altitude from the LBD_n and the distance between the projection in $\mathbb{R}^2$ of FES_n and its serving LBD_n at each time-slot t, respectively. For simplicity, at the loading process, the position of each FES_n and its corresponding LBD_n are assumed to be known. We consider a set of all possible positions $\mathcal{D}$ for each FES_n during the loading process:

$$\mathcal{L} = \{[x_n^{LBD_n}, y_n^{LBD_n}, h_n^{LBD_n} \in \mathcal{D}]\} | n = 1, \ldots, N \quad (2)$$

The received signal strength at FES_n from LBD_n at each time-slot t can be derived based on the FSO range equation [42] as follows:

$$P_n^{LBD_n}[t] = \psi \Upsilon_{LBD_n} \frac{O.EF_0 e^{-\theta d_n^{LBD_n}[t]}}{(Sz + d_n^{LBD_n}[t]\Delta\beta)^2} \quad (3)$$

where Sz is the size of the laser beam, O is the receiver telescope's area of FES_n, EF_0 is the efficiency of the transmission receiver optical, and θ is the medium attenuation's coefficient in m^{-1}. $\Delta\beta$ is the laser beam's angular spread that can be estimated as $\frac{S_d}{F_l}$, where S_d and F_l are the detector size and its focal length, respectively. In this work, we suppose that the laser power transmission follows a linear energy harvesting model with a constant efficiency $\Phi \in (0, 1)$. Therefore, the concrete harvested energy at FES_n at time-slot t is given by:

$$E_n^{LBD_n}[t] = \Phi P_n^{LBD_n}[t] \quad (4)$$

It is worthy to note that (3) and (4) are verified under a clear weather condition, *i.e.,* the value of θ tends to 10^{-7}m. Consequently, the variations of $E_n^{LBD_n}[t]$ over the distance $d_n^{LBD_n}[t]$ are dominated by $(Sz + d_n^{LBD_n}[t]\Delta\beta)^{-2}$ in this case. Moreover, notice that Υ_k is supposed to be large and $\Delta\beta$ is considered to be very small, as in [43] (*i.e.,* $\Upsilon_{LBD_n} = 1$kw and $\Delta\beta = 3.4 \times 10^{-5}$). As a result, $E_n^{LBD_n}[t]$ generally diminishes much more slowly over the distance $d_n^{LBD_n}[t]$ than it does over RF energy harvesting [44]. This also shows that the laser power transmission could have a much longer charging distance to satisfy the energy needs of FESs. However, in this work and to increase the efficiency of the laser-powered

transmission, FESs should be in close proximity to LBDs during the loading process.

As for UAV charging, laser beams are not considered due to two reasons. First, it is assumed that the BS is located at the edge of the target area (*i.e.,* far from UAVs) and cannot ensure LoS with UAVs. Second, even if Laser beams can be generated from FESs, there would be an excessive energy consumption of FESs and can hinder the proper functioning of this strategy. As a solution, we assume that in the FES-to-UAV channel, each FES is equipped with A antennas generating multiple beams, each of which is assigned with a unique identifier and covers a certain direction without overlapping (*c.f.,* Fig. 4).

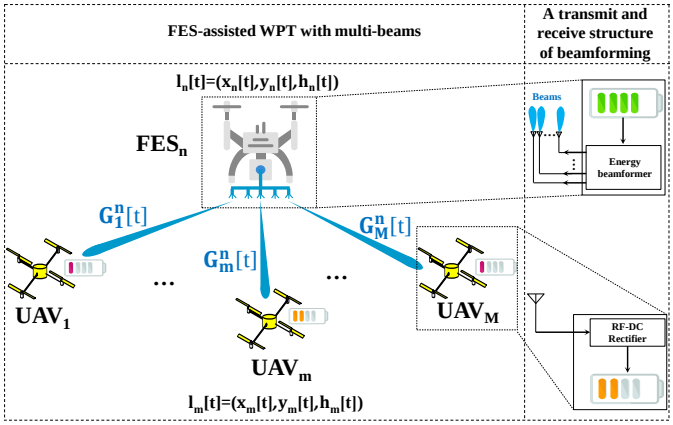


Fig. 4: Channel modeling of FES-enabled charging process.

Each beam transmits an RF-signal to a UAV that is equipped with a pair of antennas operating over orthogonal frequency bands, each of which is devoted to either data transmission or energy harvesting to avoid the problem of interference. After intercepting the RF-signals from FESs, the UAV processes the signals, transforms them into direct current (DC) energy, and then stores it in its embedded battery. Since FESs have a certain altitude and serve other flying UAVs, we assume that FES-to-UAV channel is LoS dominated with a path loss exponent $\alpha \in [2, 4]$. It should be stressed that the Doppler effect caused by the FES mobility is assumed to be perfectly compensated at the UAV receivers based on the GPS, where the velocities and positions of FESs can be accurately predicted. Therefore, the time-varying channel between a given FES$_n$ and UAV$_m$ can be expressed as follows [14]:

$$CH_m^n[t] = \sqrt{\upsilon_0 d_m^n[t]^{-\alpha}} b(\Omega) \tag{5}$$

where υ_0 denotes the power gain at the reference distance $d_0 =$ 1m. $b(\Omega)$ represents a vector of the time difference between each antenna element, which provides the angle the normal direction of the array and the beam direction. By considering a uniform linear array, $b(\Omega)$ can be formulated as follows:

$$b(\Omega) = [1, \ldots, e^{j\frac{2\pi a g}{\lambda}\sin\Omega}, \ldots, e^{j\frac{2\pi(A-1)g}{\lambda}\sin\Omega}]^T \tag{6}$$

where λ is the wavelength, which is equal to $\frac{v}{q}$, where v is the speed of the light and q is the carrier frequency. g denotes the spacing between the antenna elements. $a \in \{0, \ldots, A-1\}$ represents the coordinate of the ath antenna element. Therefore, the effective time-varying channel gain between FES$_n$ and UAV$_m$ is expressed as:

$$G_m^n[t] = \frac{\mu_0}{(d_m^n[t])^{\frac{\alpha}{2}}} |b^H(\Omega)U|^2 \tag{7}$$

where $U = [u_0, \ldots, u_a, \ldots, u_{A-1}]$ denotes the beamforming vector that describes the phase and amplitude excitation of each array element, *i.e.,* $u_a = AE_a(\Omega) I_a e^{-j\frac{2\pi}{\lambda} a g \sin\Omega}$, where I_a and $AE_a(\Omega)$ are the amplitude excitation and the element pattern of the ath array element, respectively. In the case when the embedded UAVs' batteries have unrestricted capacity, the total harvested energy at UAV$_m$ from all FESs at time-slot t is given by:

$$E_m(\{l_n[t], l_m[t]\}, t) = \sum_{n=1}^{N} E_m^n[t] = \sum_{n=1}^{N} \xi G_m^n[t] \Upsilon_n \psi \tag{8}$$

where $0 < \xi \leq 1$ denotes the RF-to-DC energy conversion efficiency at each UAV$_m$. Υ_n represents the transmission power of FES$_n$. Note that many factors could affect the sum-energy received by all UAVs, such as the density of UAVs, the trajectory, and the velocity of FESs.

B. Energy Consumption

Since it is assumed that both FESs and UAVs have a limited energy supply and continuously moving, we focus here exclusively on their energy consumption. Initially, UAVs with fully charged batteries are dispatched to collect data from sensors installed at fixed locations on the ground. Moreover, FESs are deployed above UAVs to adapt to their dynamics and efficiently supply them with energy and maintain them in flight as long as possible. It is supposed that both UAVs and FESs fly with a maximum speed of $V_\vartheta^{max}, \vartheta \in \mathcal{M} \cup \mathcal{N}$, and only hovering when UAVs communicate with sensors. It should be stressed that the energy consumption dedicated to wireless communications is relatively small, and therefore it is omitted in this work. To estimate the energy consumption of each FES and UAV, we follow the model proposed in [45]. Even though this model has not been fully validated by flight experiments and neglects many important factors, such as wind speed, 3D mobility, and acceleration, it remains the most popular model in the literature because of its simplicity. It can be formulated as follows:

$$\begin{aligned} Prop(V) = & \underbrace{\varphi_b \left(1 + \frac{3V^2}{V_{tip}^2}\right)}_{blade\ profile\ power} \\ & + \underbrace{\varphi_i \left(\sqrt{1 + \frac{V^4}{4\iota_0^2}} - \frac{V^2}{2\iota_0^2}\right)}_{induced\ power} + \underbrace{\frac{1}{2}\delta \Xi \zeta \kappa V^3}_{parasite\ power} \end{aligned} \tag{9}$$

where φ_b represents the induced power in a hover state and the blade profile power, respectively. V denotes the flying speed of both FES and UAV, V_{tip} represents the tip speed of the rotor blade, and ι_0 and ζ indicate the mean induced velocity and the

solidity of the rotor, respectively. δ, Ξ, and κ are the fuselage drag ratio, the air density, and the rotor disc area, respectively. However, there is an exception for FESs, where their energy consumption is also related to the energy transmission from their own batteries towards UAVs. Indeed, each FES$_n$ have two modes of functionality, where $\varrho_n[t] = 1$ means that FES$_n$ is transferring energy towards UAVs, otherwise, $\varrho_n[t] = 0$. As a consequence, the energy consumption of each UAV$_m$ and FES$_n$ until the current time-slot t can be estimated as follows:

$$C_n(\{\varrho_n[t], l_n[t]\}, t) = \underbrace{\int_0^t Prop(\|v_n[t]\|)dt}_{propulsion\ energy} + \underbrace{\int_0^t \varrho_n[t]\Upsilon_n dt}_{energy\ transfer} \tag{10a}$$

$$C_m(\{l_m[t]\}, t) = \underbrace{\int_{t-1}^t Prop(\|v_m[t]\|)dt}_{propulsion\ energy} \tag{10b}$$

where $v_n(t) = \dot{l}_n(t)$ and $v_m(t) = \dot{l}_m(t)$ represent the velocities of FES$_n$ and UAV$_m$, respectively. To be more realistic, we suppose that the battery of each UAV$_m$ has a capacity limited to E_m^{max}. Thus, at time-slot t, the residual energy of UAV$_m$ after harvesting energy from the FESs can be estimated as follows:

$$E_m[t] = \begin{cases} E_m^{max}, & \text{if } E_m[t-1] + \sum_{n=1}^{N} E_m^n[t-1] \geq E_m^{max} \\ E_m[t-1] + \sum_{n=1}^{N} E_m^n[t-1], & \text{Otherwise} \end{cases} \tag{11}$$

Therefore, the residual energy of each FES$_n$ and UAV$_m$ at each time-slot $t+1$ is expressed as follows:

$$E_n[t+1] = E_n^{max} - C_n(\{l_n[t], p_n[t]\}, t), \quad \forall n \in \mathcal{N} \tag{12a}$$

$$E_m[t+1] = E_m[t] - C_m(\{p_m[t]\}, t), \quad \forall m \in \mathcal{M} \tag{12b}$$

where E_n^{max} is the maximum energy capacity of each FES$_n$. At each time-slot t, the battery levels of each *active* UAV$_m$ and FES$_n$ are supposed to be discretized into 5 different levels, namely energy levels and denoted by $EL_\vartheta[t] \in \{0, 1, 2, 3, 4\}, \forall \vartheta \in \mathcal{M} \cup \mathcal{N}$. The ratio of the residual energy $E_\vartheta[t]$ of each UAV and FES can be classified into energy levels based on the following equation:

$$EL_\vartheta[t] = \left\lfloor 4 - \frac{E_\vartheta^{max} - E_\vartheta[t]}{E_\vartheta^{max}\tau} \right\rfloor, \forall \vartheta \in \mathcal{M} \cup \mathcal{N} \tag{13}$$

where $\lfloor . \rfloor$ is the floor function and τ is the threshold that is assumed to be set to 20% in which a given UAV$_m$ or FES$_n$ transits between energy levels (*e.g.,* UAV$_m$ could pass into "*critical*" state if $\frac{E_m[t]}{E_m^{max}} < \tau$). Moreover, it is supposed that τ is the minimum threshold to allow FESs to reach the BS for the loading process. FESs should continually move in each time-slot to serve some UAVs despite others while minimizing their energy consumption. For this purpose, we define a priority Ψ, where each UAV$_m$ has its own priority $\Psi_m[t] \in \{0, 1, 2, 3, 4\}$ that is calculated as $\Psi_m[t] = 4 - EL_m[t]$. The highest priority is set to 4, which indicates that UAV$_m$ has an energy level $0\% < \frac{E_m[t]}{E_m^{max}} \leq 20\%$ at time-slot t. We favor to serve active UAVs with high priorities, which are more likely to deplete their batteries quickly and falling to the ground. We introduce a binary variable $Ac_\vartheta[t] \in \{0, 1\}, \forall \vartheta \in \mathcal{M} \cup \mathcal{N}$, which takes the value of 1 if a given UAV$_m$ or FES$_n$ is active ($E_\vartheta[t] > 0, \forall \vartheta \in \mathcal{M} \cup \mathcal{N}$) and 0 otherwise ($E_\vartheta[t] = 0, \forall \vartheta \in \mathcal{M} \cup \mathcal{N}$). Another binary variable $Nb_m[t] \in \{0, 1\}$, $\forall m \in \mathcal{M}$, is also considered to define if an active UAV$_m$ has a fully charged battery or not. $Nb_m[t]$ takes the value of 0 if UAV$_m$ has a fully charged battery or inactive, and 1 otherwise in a time-slot t.

$$Nb_m[t] = \begin{cases} 0, & \text{if } E_m[t] = E_m^{max} \vee Ac_m[t] = 0 \\ 1, & \text{Otherwise} \end{cases} \tag{14}$$

The total number of active UAVs in which their batteries could be charged is given as follows:

$$UC[t] = \sum_{m=1}^{M} Nb_m[t] \tag{15}$$

To have a clear idea of how the energy levels of UAVs that are not fully charged are distributed around the average μ (*i.e.,* average of energy levels of active UAVs with not fully charged batteries). Generally, a large standard deviation means that energy levels are more dispersed around the average. In contrast, a small standard deviation indicates that the energy levels are not widely dispersed around the average. Therefore, σ^t is calculated as follows:

$$\sigma^t = \sqrt{\frac{\sum_{m=1}^{M} ((EL_m[t] \times Nb_m[t]) - \mu)}{UC[t]}} \tag{16}$$

C. Problem Statement

In line with the preceding discussions, and before we proceed with the problem statement, we present some assumptions and concepts, which are required for the proper functionality of the proposed approach. FESs and UAVs are supposed to be uniformly distributed over the target area. Initially, we assume that both UAVs and FESs are fully charged, and their energy capacities are finite and set to E_n^{max} and E_m^{max}, respectively. Our main objective is to find a control policy defining how FESs should serve UAVs in each time-slot and fly back to BS for the loading process when it is needed. To mathematically formulate the problem, we first define the charging service status of UAV$_m$ with FES$_n$ at a time-slot t is given by:

$$SER_m^n[t] = \begin{cases} 1, & \text{if } E_m^n[t] > 0 \wedge Nb_m[t] = 1 \wedge \Psi_m[t] > 0 \\ 0, & \text{Otherwise} \end{cases} \tag{17}$$

To estimate the relative FES service load of UAV$_m$ at time-slot t, we define $LD_m[t] \in \{0, 1\}$, which is calculated as follows:

$$LD_m[t] = \frac{\sum_{n=1}^{N} SER_m^n[t]}{N} \tag{18}$$

We also define $PR_n^m[t] \in \{0, 1, 2, 3, 4\}, \forall n \in \mathcal{N}, \forall m \in \mathcal{M}$, to indicate the energy transfer priority of FES$_n$ to UAV$_m$ at a time-slot t, which is expressed as follows:

$$PR_n^m[t] = \begin{cases} \Psi_m[t], & \text{if } E_m^n[t] > 0 \wedge Nb_m[t] = 1 \\ 0, & \text{Otherwise} \end{cases} \quad (19)$$

The total energy transfer priority of a given FES$_n$ at time-slot t is given by (20):

$$PR_n[t] = \sum_{m=1}^{M} PR_n^m[t] \quad (20)$$

The total energy transfer priorities of all FESs at each time-slot t is given by (21):

$$PR[t] = \sum_{n=1}^{N} \sum_{m=1}^{M} PR_n^m[t] \quad (21)$$

The proposed approach aims to maximize the harvested energy among UAVs with a low energy level (or with the highest priorities). However, it is noticed that the energy levels of UAVs are dynamic and can change over time. Therefore, some UAVs with medium and high energy levels may never be serviced, which leads to unfair charging services. As a solution, our objective is to balance the service of FESs over UAVs in a fair manner at each time-slot t. To do so, we use the widely-known metric of fairness, namely the Jain's fairness index [46]. The fairness index up to a time-slot t is given by:

$$F[t] = \frac{\left(\sum_{m=1}^{M} LD_m[t]\right)^2}{UC[t]\left(\sum_{m=1}^{M}(LD_m[t])^2\right)} \quad (22)$$

We formulate an optimization problem for our scenario to provide efficient and fair energy transfer to UAVs while jointly minimizing the energy consumption of FESs, maintaining all UAVs active, and ensuring that all FESs remain active for a long time-averaged term. This optimization mainly depends on the total energy transfer priorities of all FESs $PR[t]$, fairness level $F[t]$, the activity status of each UAV$_m$ $Ac_m[t]$, and energy level $EL_n[t]$ of each FES$_n$. For convenience, let $L = \{l_n[t], \forall n \in \mathcal{N}\}$ be the set of positions of FESs at each time-slot t. Thus, we propose addressing the following optimization problem:

$$\mathcal{P}: \max_{L} \quad \mathbb{E}\left[\frac{\sum_{t=1}^{T}(F[t]PR[t])}{\sum_{t=1}^{T}(\sigma^t(M-\sum_{m=1}^{M}Ac_m[t]))+1} \times \frac{\sum_{t=1}^{T}\sum_{t=1}^{N}EL_n[t]}{N\times T}\right]$$

$$\begin{aligned}
\text{s.t.} \quad & \textbf{C1:} \quad Ac_\vartheta[t] \in \{0,1\}, \quad \forall \vartheta \in \mathcal{M} \cup \mathcal{N}, \forall t \in \mathcal{T} \\
& \textbf{C2:} \quad \sum_{n=1}^{N} Ac_n[t] = N, \quad \forall t \in \mathcal{T} \\
& \textbf{C3:} \quad \sum_{m=1}^{M} Ac_m[t] = M, \quad \forall t \in \mathcal{T} \\
& \textbf{C4:} \quad d_n^c[t] > Sa, \quad \forall n \neq c \in \mathcal{N}, \forall t \in \mathcal{T} \\
& \textbf{C5:} \quad d_n^{BS}[t] > Sa, \quad \forall n \in \mathcal{N}, \forall t \in \mathcal{T} \\
& \textbf{C6:} \quad 0 \leq x_n[t] \leq w, \quad \forall n \in \mathcal{N}, \forall t \in \mathcal{T} \\
& \textbf{C7:} \quad 0 \leq y_n[t] \leq w, \quad \forall n \in \mathcal{N}, \forall t \in \mathcal{T} \\
& \textbf{C8:} \quad A1 < h_n[t] \leq A2, \quad \forall n \in \mathcal{N}, \forall t \in \mathcal{T}
\end{aligned} \quad (23)$$

where $\mathbb{E}[.]$ is calculated by considering the randomness of UAV mobility. Constraint **C1** represents the activity status of UAVs and FESs. **C2** and **C3** ensures that the number of FESs and UAVs initially deployed, remains all active at each time-slot t. **C4** and **C5** denote that FESs should respect the safety distance Sa between each other and with the BS at each time-slot t to avoid collisions. **C6** and **C7** guarantees that FESs will not cross the boundaries of the target area during the whole flight mission. **C8** restricts FESs to fly in its dedicated flight space $[A1, A2]$ to avoid collisions with UAVs. It is complex to address all these constraints based on existing optimization techniques. This is because it is impractical to explore the dynamic movement of UAVs, and thus it is challenging to adapt to the dynamic changes of the environment. Moreover, it is distinguished that $\mathcal{P}$ (23) is a mixed-integer non-linear program (MINLP) due to the existence of both a binary variable $Ac_\vartheta[t]$ and continuous variables $h_n[t]$, $x_n[t]$, and $y_n[t]$, which is generally computationally complex to solve it efficiently, and especially for large-scale networks. To optimally address this issue at a low complexity, we develop a deep reinforcement learning algorithm to maximize the harvested energy at UAVs fairly.

IV. ENGINE: A Deep Reinforcement Learning Method

In RL, an agent learns how to optimally interact (*i.e.,* taking a series of actions $\mathcal{A}$) with an environment system $\mathcal{S}$, to maximize a numerical reward. At each time-slot t, the agent observes the state $s_t \in \mathcal{S}$ and executes an action $a_t \in \mathcal{A}$ based on the policy $\pi(s_t, a_t)$, which updates s_t to $s_{t+1} \in \mathcal{S}$. This process is repeated until the end of the episode. The tuple, (s_t, a_t, r_t, s_{t+1}), is exploited repeatedly to enhance the policy π until the policy converges to an optimal policy. Nevertheless, RL is inappropriate for complex environments characterized by continuous action spaces and high dimensional state spaces. To improve the RL algorithms' learning speed and their performances, the DRL exploits the advantage of a deep neural network (DNN) to train the learning process.

Deep Deterministic Policy Gradient (DDPG) is a widely adopted DRL algorithm for continuous control problems [47]. DDPG is based on two DNNs, namely actor and critic (AC) networks, where the actor network is represented as $\pi(s|\eta^\pi)$, which denotes the current policy by obtaining optimal actions a_t based on specific states s_t. The critic network is, however, represented as a deep Q network $Q(s, a|\eta^Q)$ in which its parameter is learned based on the Bellman equation as in Q-learning. The actor-network $\pi(s|\eta^\pi)$ can be updated by:

$$\nabla_{\eta^\pi} J(\eta^\pi) \approx \mathbb{E}\left[\nabla_{\eta^\pi}\pi(s|\eta^\pi)|_{s=s_t} \nabla_a Q(s, a|\eta^Q)|_{s=s_t, a=\pi(s_t)}\right] \quad (24)$$

It is worth noting that the actor and critic networks apply the experience reply and target network to ensure convergence and improve performance. However, the movement optimization problem (23) is very challenging since it needs to jointly optimize the FES trajectory and energy transfer priorities. To address this problem, we adopted the use of the observable Markov game approach based on a multi-agent architecture [48]. The choice of this kind of architecture is based on the fact that it has an excellent exploration capacity and can

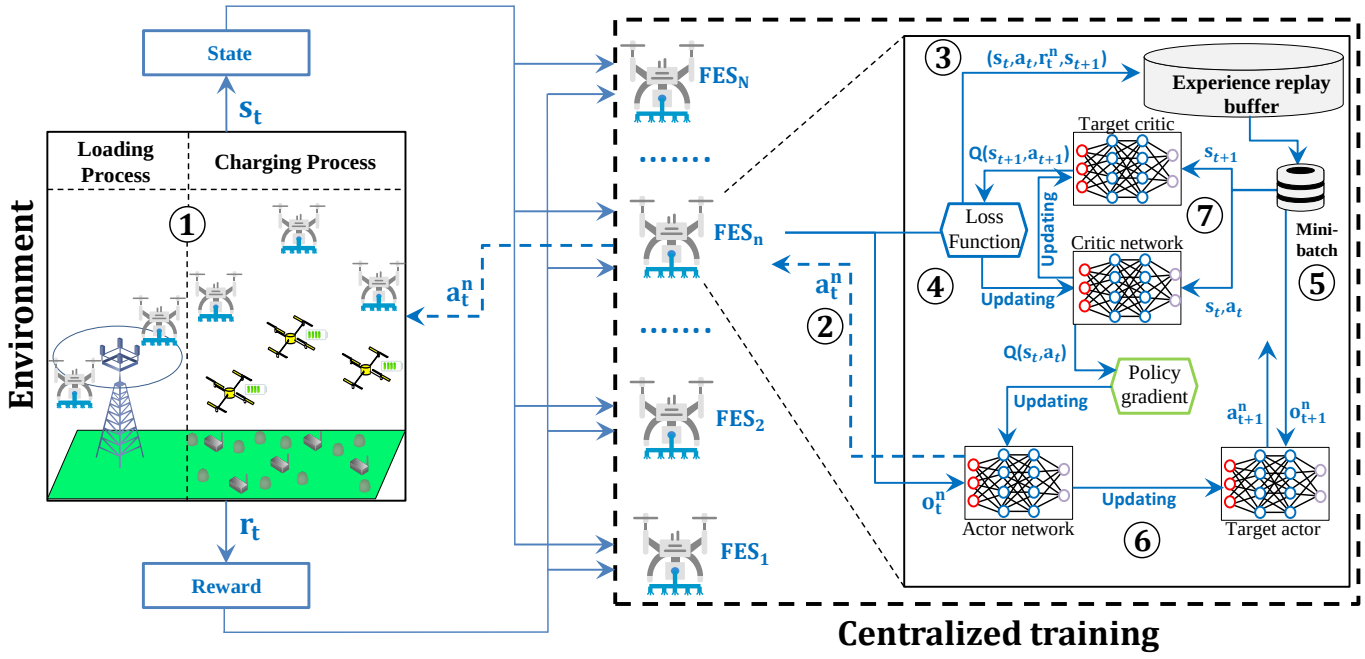


Fig. 5: Structure of the FES agent.

efficiently find an optimized solution by learning, particularly in continuous action space. Moreover, since we will deploy as many agents as there are FESs, the DDPG is adopted by each agent to address its corresponding decision process. Furthermore, this architecture centralizes the learning and distributes the execution using other agents' observations and actions.

A. ENGINE Description

There are N agents deployed at the BS representing the N deployed FESs, which are computational processes able to perform decisions, timely respond, and follow up on goals. Each agent interacts with its corresponding FES in a sequence of actions that are selected from the feasible continuous actions $\mathcal{A} \triangleq \{a_t, t \in \mathcal{T}\}$. Also, the agents interact with the dynamic UAV environment characterized by a set of states $\mathcal{S} \triangleq \{s_t, t \in \mathcal{T}\}$. The states are denoted as $s_t = \{o_t^n, t \in \mathcal{T}, n \in \mathcal{N}\}$, where o_t^n is the private observation of FES_n. The actions are defined as $a_t = \{a_t^n, t \in \mathcal{T}, n \in \mathcal{N}\}$, where a_t^n is the action taken by FES_n. At each time-slot t, each agent observes its private observation o_t^n, takes its own action a_t^n, and receives its own reward r_t^n. Then, the UAV environment updates the current state s_t and passes it to a new state s_{t+1}. It is worth noting that each agent $n, n \in \mathcal{N}$ maintains a buffer B^n, a critic network $Q^n(s_t, a_t|\eta^{Q^n})$, an actor network $a_t^n = \pi^n(o_t^n|\eta^{\pi^n})$, and their respective target networks $Q^{n'}(s_{t+1}, a_{t+1}|\eta^{Q^n})$ and $a_{t+1}^{n'} = \pi^{n'}(o_{t+1}^n|\eta^{\pi^{n'}})$. As depicted in Fig. 5, the optimization of our approach is conducted based on a centralized training combined with a distributed training. Each agent n calculates and sends its own action a_t^n to the environment, and then it receives both its own reward r_t^n and the private observations of all other agents $s_t = \{o_t^n, t \in \mathcal{T}, n \in \mathcal{N}\}$. It should be stressed that all the agents can share their own private information with the BS, such as observations and actions. This information is exploited by the critic network to be trained, which allows each agent n to estimate its own action a_t^n using only its private observation o_t^n. The primary objective of each agent n is to maximize the accumulated rewards. Therefore, all FESs should operate in an intelligent, orderly, and fair way to achieve the maximum charging service on the UAV network.

The advantage of ENGINE is that it allows performing centralized training and distributed execution, where each agent learns its state-action value function separately. In addition, each agent can get its action without knowing all information about the other agents. Another advantage is that ENGINE is based on the actor and critic networks to stabilize the learning phase. Furthermore, the critic network generates state-action values for each agent based on observations and actions from all agents, which can be used to evaluate the policy performance during the training phase. However, it is also fair to recognize that ENGINE does not behave well in an environment with a large number of agents. This is due to the fact that with the increasing number of agents in the system, it adds further complexity to the critic network when dealing with the input with larger dimensions. As a result, the critic network may become slower or even show problematic convergence, thus impacting the actor network's training speed. In the following, we describe the three basic components of our proposed model ENGINE.

1) State space: Each agent n observes the UAV environment state $s_t = \{o_t^n, t \in \mathcal{T}, n \in \mathcal{N}\}$. Each observation o_t^n includes the following details:

- $l_m[t] = [x_m[t], y_m[t], h_m[t]]$, $\forall m \in \mathcal{M}$: the current positions of all UAVs.
- $E_m[t]$, $\forall m \in \mathcal{M}$: the residual energy of all UAVs.
- $l_n[t] = [x_n[t], y_n[t], h_n[t]]$, $n \in \mathcal{N}$: the current position of FES_n.

- $E_n[t]$, $n \in \mathcal{N}$: the current residual energy of FES$_n$.
- $LD_m[t]$, $\forall m \in \mathcal{M}$: The FES service loads of all UAVs.
- $S_n[t] \in \{0,1\}$, $n \in \mathcal{N}$: takes the value of 0 if an active FES$_n$ is at the loading process ($l_n[t] \in \mathcal{L}$), and 1 otherwise (*i.e.,* FES$_n$ is performing the charging process of UAVs).

At each time-slot t, the format of the observation o_t^n is defined as $o_t^n = [l_1[t], \ldots, l_M[t], E_1[t], \ldots, E_M[t], LD_1[t]$ $, \ldots, LD_M[t]^t, l_n[t], E_n[t], S_n[t]]$, with a cardinality of $3M + 3$. Since all FESs and UAVs are supposed to be backhaul connected to the BS, in the online testing phase, each of these variables is collected by the BS at each time-slot and serves as an input of ENGINE. It should be stressed that all elements of o_t^n are normalized to accelerate the learning process. In detail, all elements that can take values greater than 1 are divided to their maximum corresponding values.

2) Action space: The action space is defined as $\mathcal{A} = \{\mathcal{A}_c(t), \mathcal{A}_l(t), \mathcal{A}_{tl}(t)\}$, where $\mathcal{A}_c(t) = \{\omega_n[t], h_n[t], d_n[t]\}$ is the actions that could be taken during the charging process and they consist of three parts defined as follows:

- $\omega_n[t] \in [0, 2\pi[$: the azimuthal angle of the FES$_n$ or its horizontal flying direction.
- $h_n[t] \in]A1, A2]$: the interval altitudes of FES$_n$.
- $d_n[t] \in [0, d_{max}]$: the flying distance of FES$_n$. If $d_n[t] = 0$, it that FES$_n$ is hovering at the same location. Otherwise, FES$_n$ moves to a certain distance $d_n[t]$ with a fixed velocity $V \in [0, V_n^{max}]$.

The actions of FES$_n$ during the loading process are limited to $\mathcal{A}_l(t) = \{\omega_n[t], h_n[t], d_n[t]\}$, where $\omega_n[t] = 0$, $d_n[t] = 0$, and no new altitude is selected $h_n[t] = 0$, which means that FES$_n$ is performing the hovering action until the loading is achieved. The action taken by FES$_n$ when it needs to make the loading process is restricted to $\mathcal{A}_{tl}(t) = \{\omega_n[t], h_n[t], d_n[t]\}$, where $\omega_n[t] = \arccos\left(\frac{x_n[t]x_n^{LBD_n} + y_n[t]y_n^{LBD_n}}{\sqrt{(x_n^2[t]+y_n^2)((x_n^{LBD_n})^2+(y_n^{LBD_n})^2)}}\right)$, $h_n[t] = h_n^{LBD_n}$, and $d_n[t] = \sqrt{(x_n[t] - x_n^{LBD_n})^2 + (y_n[t] - y_n^{LBD_n})^2}$. It should be stressed that this action must not be selected by $\mathcal{A}_c(t)$. Consequently, the selection of the different actions $a_t^n \in \tilde{\mathcal{A}}(t)$ of FES$_n$ is given by:

$$\tilde{\mathcal{A}}(t) = \begin{cases} \mathcal{A}_l(t), & \text{if } S_n[t] = 0 \wedge E_n[t] < E_n^{max} \\ \mathcal{A}_{tl}(t), & \text{if } S_n[t] = 1 \wedge E_n[t] < E_n^{max} \\ \mathcal{A}_c(t) \setminus \mathcal{A}_{tl}(t), & \text{if } S_n[t] = 1 \wedge E_n[t] \leq E_n^{max} \end{cases} \tag{25}$$

3) Reward function: At each slot t, we reward each FES$_n$ based on the priorities of energy transfer, the current state s_t, and the next state of the environment s_{t+1}. In our proposed method ENGINE, the reward function maximizes the priorities of energy transfer in a fair way provided by FES$_n$ while maintaining all UAVs active and within an acceptable energy level during the whole mission. The reward function is defined as follows:

$$r_t^n = EL_n[t] \times \frac{F[t]PR_n[t]}{\sigma^t(M - \sum_{m=1}^{M} Ac_m[t]) + 1} \tag{26}$$

Generally speaking, the reward function focuses on fairly distributing energy among UAVs while maintaining their energy levels at about the same level to avoid their failures. We punish all FESs for flying out of the target area and collisions between each other or with the BS. Moreover, a penalty is given at each FES$_n$ for making the loading process. Consequently, three penalties are incurred by each FES$_n$:

$$\rho_n^1[t] = \begin{cases} \Lambda_1, & \text{if } x_n[t] \wedge y_n[t] \notin [0, w], \\ \Lambda_1, & \text{if } h_n[t] \notin]A1, A2], \\ 0, & \text{Otherwise} \end{cases} \tag{27}$$

$$\rho_n^2[t] = \begin{cases} \Lambda_2, & \text{if } d_n^{BS}[t] \leq Sa, \\ \Lambda_2, & \text{if } d_n^c[t] \leq Sa, \forall n \neq c \in \mathcal{N} \\ 0, & \text{Otherwise} \end{cases} \tag{28}$$

$$\rho_n^3[t] = \begin{cases} 0, & \text{if } S_n[t] = 1, \\ \Lambda_3, & \text{Otherwise} \end{cases} \tag{29}$$

The penalties Λ_1, Λ_2, and Λ_3 are incurred by each FES$_n$, whenever an action a_t^n would result in violating the safety distance Sa, crossing the target area, or selecting the loading process, respectively. Then, at each time-slot t, all penalties are summed for each FES$_n$, *i.e.,* $\rho_n[t] = \sum_{i=1}^{3} \rho_n^i[t]$, and incurred from its corresponding reward r_t^n.

B. ENGINE Algorithm

As depicted in Fig. 5, the ENGINE's implementation consists of the environment, the obtained reward with incurred penalties, and the different neural networks. The environment can be partially observed by each agent $n \in \mathcal{N}$, where the actor and critic networks estimate the optimal control policy of each FES$_n$.

The algorithm of ENGINE is formally presented in Algorithm 1 and illustrated in Fig. 5. This algorithm is executed by each agent n, which controls the actions of its corresponding FES$_n$ based on a DDPG algorithm and tries to find an optimal policy π^{n^*}. Initially, the reply buffer $\mathcal{B}$ of size B is initialized (Line 2). We randomly initialize the critic network $Q^n(.)$ and the actor network $\pi^n(.)$ with their respective weights η^{Q^n} and η^{π^n} (Line 3). As for Line 4, we create target networks $Q^{n'}(.)$ and $\pi^{n'}(.)$ based on the same structure as $Q(.)$ and $\pi(.)$ with their respective weights $\eta^{Q^{n'}} \leftarrow \eta^Q$ and $\eta^{\pi^{n'}} \leftarrow \eta^\pi$. In line 5, we initialize the action noise ε. It should be stressed that the parameters $\eta^{Q^{n'}}$ and $\eta^{\pi^{n'}}$ are slowly updated at the end of the algorithm (Lines 30-32) based on the parameter $\chi = 0.001$ for the sake of stability. In Lines 6-7, we initialize the number of episodes EPS and the number of epochs T.

The second part of the algorithm (Lines 9-32) represents the training process of ENGINE over EPS episodes, and in each episode, there are T time steps. In Lines 10-11, the partial observation of the environment o_t^n and location $l_n[t]$ of each FES$_n$ is randomly initialized ①. As for Lines 13-17, at each time-slot $t \in \mathcal{T}$, ENGINE selects a trajectory action a_t^n for each FES$_n$ based on the actor-network $\pi^n(o_t^n|\eta^{\pi^n})$ with an additional random noise ε that decreases over epochs with

a rate of 0.9995 ②. When an action a_t^n is executed, FES$_n$ receives a reward r_t^n and transits to the next state s_{t+1}. Then, an observation is made to see if FES$_n$ have exceeded their restrictions (*see* (27), (28), and (29)). If this is the case, a set of penalties is calculated in Line 17. These penalties are deducted from the reward r_t^n in Line 20. Moreover, the action a_t^n is canceled and the state of the environment is updated accordingly to s_{t+1} (Line18). In the case when no penalty is observed, the reward r_t^n is calculated, and a new state s_{t+1} is obtained.

In the last part of ENGINE, each agent n collects the tuple $(s_t, a_t, r_t^n, s_{t+1})$ of each training time step, which is stored in its replay buffer $\mathcal{B}_n$ ③. Then, a random mini-batch samples Γ tuples from the buffer $\mathcal{B}_n$ to make the update of the actor and critic networks based on three steps ⑤. First, the target value TGT_γ^n is calculated based on the target critic network $Q^{n'}(.)$, where ν is a discount factor ⑦. Second, the loss function $Loss(\eta^{Q^n}$ updates the critic network ④. Third, the policy gradient $\nabla_{\eta^{\pi^n}} J(\eta^{\pi^n})$ updates the actor network. The update of the parameters of the actor and critic networks based on the same method in [47].

C. Complexity and Implementation Analysis

From the view of complexity, ENGINE operates well with the increase of FESs' density. This is explained by the fact that each FES' computational complexity is only related to its neural network configuration, where the density of FESs is considered to be linear to the size of the input layers (*see* Table IV in Section V-A). The computational complexity of Algorithm 1 is mainly determined by the density of FES agents and the organization of the actor and critic networks for each agent. Let $\eth_{A,i}$ be the unit number in the ith layer of the actor network and $\eth_{C,j}$ be that in the jth layer of the critic network. Then, it is supposed that the critic and actor networks of each FES agent comprise J and I fully connected layers, respectively. If the number of agents in the system is M, the computational complexity can be calculated as follows:

$$\begin{aligned}\Im_{cplx} &= M \times \left(2 \times \sum_{i=1}^{I-1} \eth_{A,i}\eth_{A,i+1} + 2 \times \sum_{j=1}^{J-1} \eth_{C,j}\eth_{C,j+1}\right), \\ &= \mathcal{O}\left(M \times \left(\sum_{i=1}^{I-1} \eth_{A,i}\eth_{A,i+1} + \sum_{j=1}^{J-1} \eth_{C,j}\eth_{C,j+1}\right)\right)\end{aligned} \tag{30}$$

It should also be stressed that ENGINE takes more time to converge because each agent has its own actor and critic networks that are updated only once at each step. Furthermore, the architecture of ENGINE has a high number of parameters due to the presence of numerous actor and critic networks. Also, the agents do not have a shared replay buffer, which will significantly slow down again ENGINE to converge during the learning phase.

D. Convergence and Communication Cost

ENGINE adopts a gradient descent method to train actor π^n and critic Q^n networks of each agent so that to update their weights η^{π^n} and η^{Q^n}, respectively, while decaying the learning rates with iterations. After a finite number of iterations, the weights will converge to particular values that guarantee the convergence of ENGINE. According to [49] and [50], the theoretical convergence analysis is very complicated to be made before network training. Instead, the convergence of ENGINE can be observed by simulations in Section V-A.

As for the communication cost among agents, the interaction of each agent with the environment generates training samples that are fed back to the BS, where a centralized training is made using all information received from the agents. This does not involve any communication between agents, and therefore ENGINE incurs no communication cost among agents.

Algorithm 1: ENGINE pseudo-code.

1 **begin**
2 Initialize replay buffer $\mathcal{B}_n$ to capacity B, where ($\mathcal{B}_n = \emptyset$);
3 Randomly initialize actor network $\pi^n(.)$ and critic network $Q^n(.)$ with their respective weights η^{π^n} and η^{Q^n};
4 Initialize target networks $\pi^{n'}(.)$ and $Q^{n'}(.)$ with weights $\eta^{\pi^{n'}} \leftarrow \eta^{\pi^n}$ and $\eta^{Q^{n'}} \leftarrow \eta^{Q^n}$;
5 Initialize the action noise ε;
6 $EPS \leftarrow$ Number of episodes;
7 $T \leftarrow$ Number of time steps;
9 **for** $Episode \leftarrow 0, \ldots, EPS$ **do**
10 Initialize $l_n[t]$ of FES$_n$;
11 Initialize state o_0^n, $\forall n \in \mathcal{N}$, $\forall m \in \mathcal{M}$;
`// All the components of` o_t^n `are initialized.`
13 **for** $t \leftarrow 0, \ldots, T$ **do**
14 $a_t^n = \pi^n(o_t^n|\eta^{\pi^n}) + \varepsilon$;
15 **Execute**: action $a_t^n = [\omega_n[t], h_n[t]d_n[t]]$, $n \in \mathcal{N}$;
17 **if** $\rho_n[t] > 0$ **then**
18 Cancel action a_t^n of FES$_n$ and update s_{t+1};
19 **Evaluate**: get reward r_t^n based on (26), $n \in \mathcal{N}$;
20 $r_t^n \leftarrow r_t^n - \rho_n[t]$;
21 **Observe**: obtain a new state s_{t+1};
22 Store transition sample $(s_t, a_t, r_t^n, s_{t+1})$ into experience buffer replay $\mathcal{B}_n$
`// Store tuples directly in the experience replay buffer`
23 Sample random mini-batch of size Γ samples of transitions $(s_\gamma, a_\gamma, r_\gamma^n, s_{\gamma+1})$ from $\mathcal{B}_n$;
24 Set target value TGT_γ^n:
25 $TGT_\gamma^n = r_\gamma^n + \nu Q^{n'}(s_{\gamma+1}, \pi^{n'}(s_{\gamma+1}|\eta^{\pi^{n'}})|\eta^{Q^{n'}})$;
26 Update weight η^{Q^n} of $Q^n(.)$ by minimizing the loss ($L(\eta^{Q^n})$):
27 $Loss(\eta^{Q^n}) = \frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma}(TGT_\gamma^n - Q^n(s_\gamma, a_\gamma|\eta^{Q^n}))^2$;
28 Update weight η^{π^n} of $\pi^n(.)$ by:
29 $\nabla_{\eta^{\pi^n}} J(\eta^{\pi^n}) \approx \frac{1}{\Gamma}\sum_{\gamma=1}^{\Gamma} \nabla_{\eta^{\pi^n}} \pi^n(o_t^n|\eta^{\pi^n}) \nabla_{a_t^n} Q^i(s_t, a_t|\eta^{Q^n})|, n \in \mathcal{N}, t \in \mathcal{T}$
30 Update the corresponding target network weights $\eta^{Q^{n'}}$ of $\eta^{\pi^{n'}}$ by:
31 $\eta^{Q^{n'}} = \chi\eta^{Q^n} + (1-\chi)\eta^{Q^{n'}}$;
32 $\eta^{\pi^{n'}} = \chi\eta^{\pi^n} + (1-\chi)\eta^{\pi^{n'}}$;

V. Performance Evaluation

In this section, we present the numerical results and performance evaluation of ENGINE. In Sub-section V-A, we provide the simulation settings and the parameters of the adopted

neural network. In Sub-section V-B, we analyze the obtained results of the training and testing stages of ENGINE, and we interpret the numerical results of other baseline methods.

A. Simulation Settings

ENGINE is trained using Tensorflow 1.14.0 and Python 3.6.9 over 2000 episodes. Each episode is divided into 100 time-slots. ENGINE is then tested over ten episodes (*i.e.,* 1000 time-slots), where the average values of the important metrics are calculated. We assume our environment to be non-flat stretching over 5000m×5000m area, and that a set $\mathcal{M}$ of UAVs is randomly moving and hovering within the target area following the Gaussian Markov model. Each UAV$_m$ can fly at an altitude $h_m[t] \in [50m, 100m]$. Another set $\mathcal{N}$ of FESs is also deployed over UAVs to provide them energy charging services. Each FES$_n$ has a flight altitude $h_n[t] \in]100m, 150m]$. Each FES$_n$ and UAV$_m$ can reach a maximum speed of $V_\vartheta^{max} = 20$m/s, $\vartheta \in \mathcal{N} \cup \mathcal{M}$. The main simulation settings are provided in Table III.

TABLE III: Simulation Setup.

Parameter	Description	Value
Surface	Area size	25 km^2
w	Area width	5 km
h_m	Altitude of each UAV$_m$	$[50m, 100m]$
h_n	Altitude of each FES$_n$	$]100m, 150m]$
UAV density	Number of UAVs	10
FES density	Number of FESs	$[2, 16]$
Υ_n	Transmission power of each FES$_n$	50 W
Υ_{LBD_n}	Transmission power of each LBD$_n$	1 kW
V_n^{max}	Maximum speed of each FES$_n$	14m/s
V_m^{max}	Maximum speed of each UAV$_m$	14m/s
μ_0	Reference channel gain	-10 dB
α	Path loss factor	2
q	Carrier frequency	700 MHz
ξ	Energy conversion	0.5
$\Delta\beta$	Laser beam's angular spread	3.4×10^{-5}
Sz	Laser beam size	0.1 m
θ	Attenuation coefficient	10^{-7}

As for the neural networks, we consider four fully connected hidden layers for both the actor and critic networks. Both networks have layers composed of 200, 200, 100, and 100 neurons, respectively, using rectified linear unit (ReLU) as an activation function. In addition, Hyperbolic tangent (tanh) is used as an activation function in the actor-network output layer to restrict movements according to the maximum travel distance of FESs. The critic network input is represented as a concatenation of observations and actions, and the output is a scalar for the evaluation of the observations according to the global policy. The parameters are listed in Table IV.

Since the dynamic of UAVs to serve is unknown for FESs, ENGINE is considered as an offline training phase, and it is mainly executed at the BS to estimate the optimal policy π^{n^*}. Then, π^{n^*} is extracted to optimize the movement of FESs to fairly serve UAVs during the online testing phase.

B. Result Analysis

The simulations are divided into two phases: (i) the training phase of ENGINE and (ii) the testing phase for a comparative study with baseline methods. To compare the performance of

TABLE IV: Parameters of ENGINE.

Parameters of actor neural network

Layers	Number	Size	Activ. functions
Input	1	$3M + 3$	–
Hidden	4	200, 200, 100, 100	ReLU
Output	1	3	Tanh

Parameters of critic neural network

Layers	Number	Size	Activ. functions
Input	1	$N(3M + 3) + 3$	–
Hidden	4	200, 200, 100, 100	ReLU
Output	1	1	–

Key parameters of the training stage

Parameter	Value
Memory size $\mathcal{B}_n$	10^5
Mini-batch size Γ	256
Actor learning rate	0.001
Critic learning rate	0.001
Optimizer method	Adam
Steps for updating target networks	1000
Reward discount, ν	0.99
ρ_n^1, ρ_n^2, ρ_n^3	10.0
RL Comparisons	DQN, Multi-Agent DQN

ENGINE, we consider other DRL methods, namely Deep Q Network (DQN) [51] and Multi-agent DQN (MADQN) [52]. Moreover, we consider two baseline methods, namely random and greedy techniques.

1) Training ENGINE: At a first step, we calculate for each episode the obtained reward and provide some analysis of the results (*c.f.,* Fig. 6). We can see that the obtained rewards increase slowly through episodes to reach peak values after 250 training episodes. This is mainly caused by the efficient learning of ENGINE to the dynamic of UAV network while making intelligent decisions to increase the energy transfer priorities among FESs.

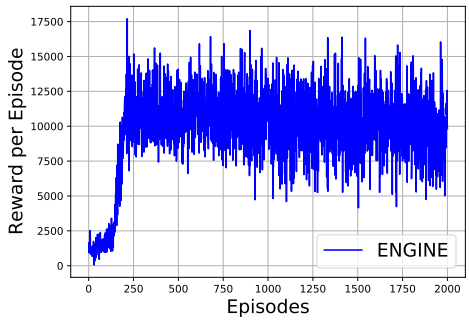
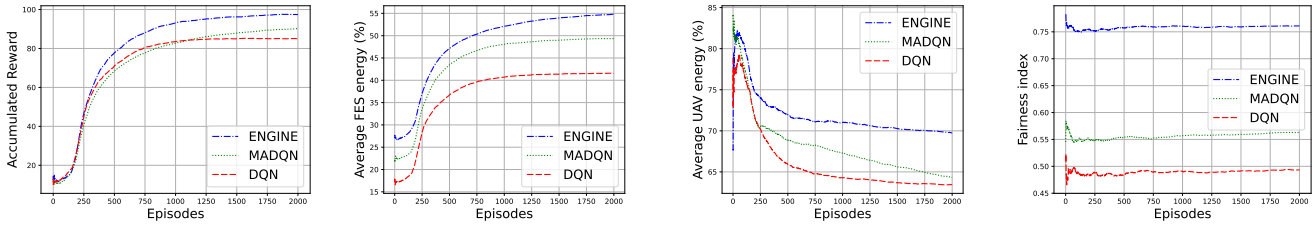


Fig. 6: Reward per episode (UAVs=10 and FESs=5).

To evaluate the convergence of ENGINE, we consider the accumulated reward, the energy level of UAVs and FESs, and the fairness index at each episode in the training process (*see* Fig. 7). For instance Fig. 7(a) shows that initially, the accumulated reward is not stable and mainly remains at a low level. Then, it continues to increase and start stabilizing after 500 episodes. This is explained by the fact that FESs are initially randomly distributed over the target area, and their priority services will be significantly decreased. Then, FESs learn how to both efficiently perform the loading/charging



(a) Cumulative reward. (b) Energy levels of FESs. (c) Energy levels of UAVs. (d) Fairness index.

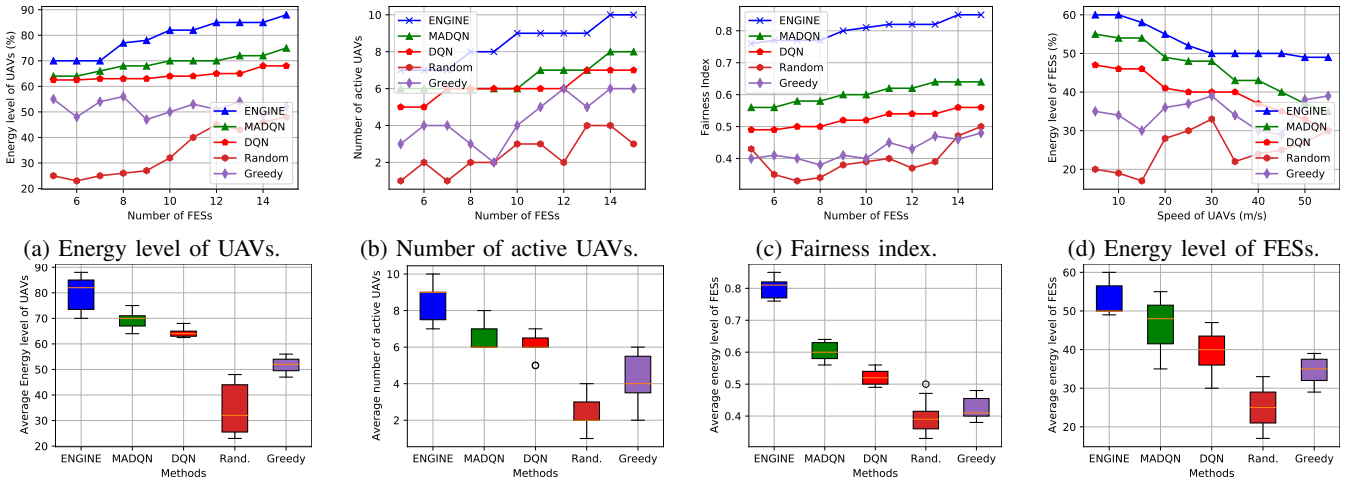
Fig. 7: Performance comparisons over episodes (UAVs=10 and FESs=5).

process and fly close to UAVs, especially those with low energy levels, resulting in an increased reward. In Fig. 7(b), we clearly distinguish that the average energy level of FESs increases quickly at the outset of the learning step. This is because FESs adapt quickly to the environment dynamic and making movements only when it is required, thus consuming less energy. At the same time, FESs avoid as much as possible serving UAVs with high energy levels and tend to maximize the energy transfer priorities to maintain the maximum number of UAVs active. From Fig. 7(c), we observe that the initial average battery level of all UAVs is between 67% and 82 % due to the energy consumed during taking off from random origins. Then, their average energy levels reach their lowest point in nearly 500 episodes but never decrease after. This is due to the continuous learning of FESs from the interaction with the dynamic environment of UAVs and builds up a good policy for multiple FESs trajectory control. Also, FESs have enhanced their trajectories to achieve time and fair energy charging services for UAVs, which results in preventing the majority of UAVs from running out of energy. Under the same number of episodes, we also trained ENGINE to calculate the fairness index as depicted in Fig. 7(d). We can clearly observe that ENGINE provides a high fairness index compared to MADQN and DQN. This is explained by the fact that FESs in ENGINE learn how to provide fairly charging services to UAVs based on a fair service policy. As for MADQN and DQN, their action space is discreet, and therefore FESs cannot provide charging services optimally and fairly to UAVs and adapt to their continuous movements.

In all these obtained results, the rewards, the average energy levels of UAVs and FESs, and the average fairness index calculated by DQN and MADQN, do not reach the optimal values compared to ENGINE and converge more slowly. This is caused by the overfitting issues of DQN and MADQN and the extracted small-batch samples with the same probability for training, which cannot distinguish the important samples.

2) Comparative study with baseline methods: In this section, we first study the impact of the FES density on the average energy levels of UAVs (*c.f.,* Fig. 8). Then, in Fig. 9, we show the average received energy by each individual UAV. From Figs. 8(a) and 8(e), two observations can be made. First, the average energy levels of UAVs in ENGINE significantly outperform those obtained by the other methods since FESs make intelligent decisions based on the knowledge of previous experiences and aim to maximize the priorities of energy transfer, and therefore increase the average energy levels of UAVs. Second, as for the random and greedy techniques, they usually have low fairness, which favors charging some UAVs in spite of others. While in DQN and MADQN, FESs cannot perform continuous actions, thus low convergence and not good performance as in ENGINE. In Figs. 8(b) and 8(f), we notice that the number of active UAVs increases continuously as the density of FESs increases. This is due to the increase of fairness in servicing UAVs, where FESs try to provide energy charging services fairly between UAVs. The number of active UAVs in DQN and MAQN has the same behavior as in ENGINE, but does not perform better. This is explained by the fact that in DQN and MAQN, some unserved UAVs tend to quickly lack energy and may fall on the ground, thus decreasing the number of active UAVs. As for the greedy and random methods, we notice a reduced number of active UAVs, which reflects the random behaviors of FESs to cover UAVs and provide the required energy supply. Figs. 8(c) and 8(g) show the fairness index according to the density of FESs. Indeed, it is not strange that the fairness index has a strong relationship with the density of FESs. Because in ENGINE, each FES tries to serve at most one UAV to increase fairness and maximize the rewards until the density of FESs exceeds the density of UAVs and the fairness reaches nearly 1. As for DQN and MADQN, even if the density of FESs exceeds the density of UAVs, fairness did not reach the same level as in ENGINE, which is due to the low convergence in these methods, and thus low fairness among UAVs. In the random and greedy techniques, we observe low and unstable fairness among UAVs due to the random movements without taking this factor into account. In Figs. 8(d) and 8(h), we draw the obtained results in terms of energy levels of FESs according to the speed of UAVs under the same number of episodes. Indeed, we distinguish that ENGINE outperforms DQN, MADQN, and the baseline methods as expected. This is because ENGINE quickly builds an optimal policy compared to the other methods, which allows a better movement control of FESs according to the speed of UAVs. Moreover, it is observed that ENGINE preserves the residual energy of FESs up to 10% better than MADQN and DQN. This is because FESs in ENGINE learn faster to place themselves in the right places where energy is needed by UAVs.

In Fig. 9, it is observed that UAVs in ENGINE receive nearly the same amount of energy, which illustrates the near-fairness performed among UAVs. On the other hand, it is



(a) Energy level of UAVs. (b) Number of active UAVs. (c) Fairness index. (d) Energy level of FESs.

(e) Average energy levels of UAVs. (f) Average nb. of active UAVs. (g) Average fairness index. (h) Energy level of FESs.

Fig. 8: FESs' density impact on (a) Energy levels of UAVs, (b) Number of active UAV, (c) fairness index, and (d) UAVs' speed impact on energy levels of FESs.

not the case for DQN and MADQN in which FESs fail to provide an acceptable fairness index among UAVs due to the low convergence of their algorithms.

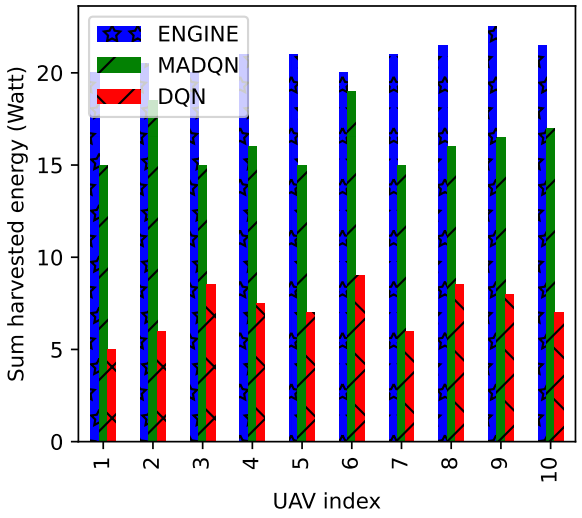


Fig. 9: Harvested energy of each UAV according to its index (FESs=5).

VI. Conclusion

This research presented a novel technique for durable and fair energy supply for UAVs using WPT technologies. This method consists of deploying flying intelligent autonomous FESs for recharging UAVs using energy beamforming WPT technology. The main benefit of this technique lies in the capacity of the FESs to distributively compute energy-efficient 3D trajectories towards the UAVs, while maintaining fairness in recharging the UAVs, allowing a maximum number of them active. In this paper, we provided detailed analytical models for various elements of our solution, including the channel model, the energy consumption model, and the beamforming-based energy harvesting model. To support our approach, we proposed using a multi-agent-based DRL method, called ENGINE, for the mobility control of the FES and the computation of a near-optimal set of corresponding trajectories. Specifically, ENGINE exploits the movements of FESs in the target 3D space, to maximize the priorities of energy transfer at each time-slot, taking into consideration the fairness among UAVs. Furthermore, the training phase of ENGINE was appropriately tuned to ensure a high level of fairness when scheduling operations of energy transfer to UAVs. This is done by maximizing the minimum power transferred to all UAVs, and therefore extending their flight time. The conducted simulations show that ENGINE with optimized FES 3D trajectory significantly improves the wireless powered UAV network performance. In addition, ENGINE outperforms baseline methods in terms of four metrics, including the average battery levels, the fairness index, the average number of active UAVs, and the average harvested energy.

However, it would be more cost-effective for the FESs to execute additional tasks to their role of flying energy sources. Therefore, for future work, we plan to extend the role of FESs in the current solution with the role of data collectors for UAVs. In this scenario, FESs can serve as mobile edges with the ability to collect data from and compute tasks for UAVs while supplying them with power. Moreover, we believe that for the real-world deployment of our solution, some physical variables, such as weather conditions, wireless interference, and the size/weight of the flying energy sources, which were neglected in the simulation, should be considered.

Acknowledgment

This project was supported in part by UAE University's National Space Science and Technology Center Project number G00003280.

REFERENCES

[1] B. Li, Z. Fei, and Y. Zhang, "UAV communications for 5G and beyond: Recent advances and future trends," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2241–2263, 2018.

[2] O. S. Oubbati, M. Atiquzzaman, P. Lorenz, M. H. Tareque, and M. S. Hossain, "Routing in flying ad hoc networks: survey, constraints, and future challenge perspectives," *IEEE Access*, vol. 7, pp. 81 057–81 105, 2019.

[3] H. Wang, H. Zhao, W. Wu, J. Xiong, D. Ma, and J. Wei, "Deployment algorithms of flying base stations: 5G and beyond with UAVs," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 009–10 027, 2019.

[4] O. S. Oubbati, A. Lakas, P. Lorenz, M. Atiquzzaman, and A. Jamalipour, "Leveraging communicating UAVs for emergency vehicle guidance in urban areas," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 2, pp. 1070–1082, 2019.

[5] X. Liu, Y. Liu, Y. Chen, and L. Hanzo, "Trajectory design and power control for multi-UAV assisted wireless networks: A machine learning approach," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7957–7969, 2019.

[6] M. Li, N. Cheng, J. Gao, Y. Wang, L. Zhao, and X. Shen, "Energy-efficient UAV-assisted mobile edge computing: Resource allocation and trajectory optimization," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3424–3438, 2020.

[7] O. S. Oubbati, M. Mozaffari, N. Chaib, P. Lorenz, M. Atiquzzaman, and A. Jamalipour, "ECaD: Energy-efficient routing in flying ad hoc networks," *International Journal of Communication Systems*, vol. 32, no. 18, p. e4156, 2019.

[8] M. T Nguyen, C. V Nguyen, L. H Truong, A. M Le, T. V Quyen, A. Masaracchia, and K. A Teague, "Electromagnetic field based WPT technologies for UAVs: A comprehensive survey," *Electronics*, vol. 9, no. 3, p. 461, 2020.

[9] S. Yin, Y. Zhao, L. Li, and F. R. Yu, "UAV-assisted cooperative communications with time-sharing information and power transfer," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 1554–1567, 2019.

[10] J. Huang, Y. Zhou, Z. Ning, and H. Gharavi, "Wireless power transfer and energy harvesting: Current status and future prospects," *IEEE wireless communications*, vol. 26, no. 4, pp. 163–169, 2019.

[11] B. Alzahrani, O. S. Oubbati, A. Barnawi, M. Atiquzzaman, and D. Alghazzawi, "UAV Assistance Paradigm: State-of-the-art in Applications and Challenges," *Journal of Network and Computer Applications*, vol. 166, p. 102706, 2020.

[12] O. S. Oubbati, M. Atiquzzaman, T. A. Ahanger, and A. Ibrahim, "Softwarization of UAV Networks: A Survey of Applications and Future Trends," *IEEE Access*, vol. 8, pp. 98 073–98 125, 2020.

[13] J. Xu, Y. Zeng, and R. Zhang, "UAV-enabled wireless power transfer: Trajectory design and energy optimization," *IEEE Transactions on Wireless Communications*, vol. 17, no. 8, pp. 5092–5106, 2018.

[14] W. Feng, N. Zhao, S. Ao, J. Tang, X. Zhang, Y. Fu, D. K. So, and K.-K. Wong, "Joint 3D trajectory design and time allocation for UAV-enabled wireless power transfer networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 9265–9278, 2020.

[15] X. Li, G. Zhu, Y. Gong, and K. Huang, "Wirelessly powered data aggregation for IoT via over-the-air function computation: Beamforming and power control," *IEEE Transactions on Wireless Communications*, vol. 18, no. 7, pp. 3437–3452, 2019.

[16] Y. Liu, H.-N. Dai, H. Wang, M. Imran, X. Wang, and M. Shoaib, "UAV-enabled data acquisition scheme with directional wireless energy transfer for Internet of Things," *Computer Communications*, vol. 155, pp. 184–196, 2020.

[17] R. Nangia, "Operations and aircraft design towards greener civil aviation using air-to-air refuelling," *The Aeronautical Journal*, vol. 110, no. 1113, pp. 705–721, 2006.

[18] Z. Na, Y. Liu, J. Shi, C. Liu, and Z. Gao, "UAV-supported Clustered NOMA for 6G-enabled Internet of Things: Trajectory Planning and Resource Allocation," *IEEE Internet of Things Journal*, vol. 8, no. 20, pp. 15 041–15 048, 2020.

[19] L. Bobaru, M. Iordache, M. Stanculescu, D. Niculae, and S. Deleanu, "Optimization Methods for Wireless Power Transfer," in *Numerical Methods for Energy Applications*. Springer, 2021, pp. 513–543.

[20] H. Xie, D. Yang, L. Xiao, and J. Lyu, "Connectivity-Aware 3D UAV Path Design with Deep Reinforcement Learning," *IEEE Transactions on Vehicular Technology*, 2021.

[21] D. Ind., "Drone market report," Tech. Rep.

[22] S. Yin, Y. Zhao, and L. Li, "Resource allocation and base station placement in cellular networks with wireless powered UAVs," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 1050–1055, 2018.

[23] A. Trotta, M. Di Felice, F. Montori, K. R. Chowdhury, and L. Bononi, "Joint coverage, connectivity, and charging strategies for distributed UAV networks," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 883–900, 2018.

[24] B. Zhang, C. H. Liu, J. Tang, Z. Xu, J. Ma, and W. Wang, "Learning-based energy-efficient data collection by unmanned vehicles in smart cities," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1666–1676, 2017.

[25] X. Liu, M. Chen, S. Wang, W. Saad, and C. Yin, "Trajectory Design for Energy Harvesting UAV Networks: A Foraging Approach," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2020, pp. 1–6.

[26] H. Huang and A. V. Savkin, "A method of optimized deployment of charging stations for drone delivery," *IEEE Transactions on Transportation Electrification*, vol. 6, no. 2, pp. 510–518, 2020.

[27] Y. Qin, M. A. Kishk, and M.-S. Alouini, "On the influence of charging stations spatial distribution on aerial wireless networks," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 3, pp. 1395–1409, 2021.

[28] L. Xie, J. Xu, and R. Zhang, "Throughput maximization for UAV-enabled wireless powered communication networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1690–1703, 2018.

[29] Y. Hu, X. Yuan, J. Xu, and A. Schmeink, "Optimal 1D trajectory design for UAV-enabled multiuser wireless power transfer," *IEEE Transactions on Communications*, vol. 67, no. 8, pp. 5674–5688, 2019.

[30] Z. Yang, W. Xu, and M. Shikh-Bahaei, "Energy efficient UAV communication with energy harvesting," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 1913–1927, 2019.

[31] F. Wu, D. Yang, L. Xiao, and L. Cuthbert, "Energy consumption and completion time tradeoff in rotary-wing UAV enabled WPCN," *IEEE Access*, vol. 7, pp. 79 617–79 635, 2019.

[32] J. Park, H. Lee, S. Eom, and I. Lee, "UAV-aided wireless powered communication networks: Trajectory optimization and resource allocation for minimum throughput maximization," *IEEE Access*, vol. 7, pp. 134 978–134 991, 2019.

[33] P. Wu, F. Xiao, C. Sha, H. Huang, and L. Sun, "Trajectory optimization for UAVs' efficient charging in wireless rechargeable sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4207–4220, 2020.

[34] T. D. P. Perera, S. Panic, D. N. K. Jayakody, P. Muthuchidambaranathan, and J. Li, "A WPT-enabled UAV-assisted condition monitoring scheme for wireless sensor networks," *IEEE Transactions On Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5112–5126, 2020.

[35] S. Ku, S. Jung, and C. Lee, "UAV Trajectory Design Based on Reinforcement Learning for Wireless Power Transfer," in *Proceedings of the 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*. IEEE, 2019, pp. 1–3.

[36] S. A. Hoseini, J. Hassan, A. Bokani, and S. S. Kanhere, "Trajectory Optimization of Flying Energy Sources using Q-Learning to Recharge Hotspot UAVs," in *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2020, pp. 1–6.

[37] J. Xu, K. Zhu, and R. Wang, "RF Aerially Charging Scheduling for UAV Fleet: A Q-Learning Approach," in *Proceedings of the 15th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*, 2019, pp. 194–199.

[38] K. Li, W. Ni, E. Tovar, and A. Jamalipour, "Deep Q-learning based resource management in UAV-assisted wireless powered IoT networks," in *Proceedings of the IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[39] J. Hassan, A. Bokani, and S. S. Kanhere, "Recharging of flying base stations using airborne rf energy sources," in *Proceedings of the IEEE Wireless Communications and Networking Conference Workshop (WCNCW)*. IEEE, 2019, pp. 1–6.

[40] L. Liu, K. Xiong, J. Cao, Y. Lu, P. Fan, and K. B. Letaief, "Average AoI Minimization in UAV-assisted Data Collection with RF Wireless Power Transfer: A Deep Reinforcement Learning Scheme," *IEEE Internet of Things Journal*, 2021.

[41] Y. Yu, J. Tang, J. Huang, X. Zhang, D. K. C. So, and K.-K. Wong, "Multi-Objective Optimization for UAV-Assisted Wireless Powered IoT Networks Based on Extended DDPG Algorithm," *IEEE Transactions on Communications*, vol. 69, no. 9, pp. 6361–6374, 2021.

[42] D. Killinger, "Free space optics for laser communication through the air," *Optics and photonics news*, vol. 13, no. 10, pp. 36–42, 2002.
[43] H. Kaushal, V. Jain, and S. Kar, *Free space optical communication*. Springer, 2017.
[44] N. Zlatanov, D. W. K. Ng, and R. Schober, "Capacity of the two-hop relay channel with wireless energy transfer from relay to source and energy transmission cost," *IEEE Transactions on Wireless Communications*, vol. 16, no. 1, pp. 647–662, 2016.
[45] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2329–2345, 2019.
[46] R. K. Jain, D.-M. W. Chiu, W. R. Hawe *et al.*, "A quantitative measure of fairness and discrimination," *Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA*, 1984.
[47] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.
[48] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Machine learning proceedings 1994*. Elsevier, 1994, pp. 157–163.
[49] F. Wu, H. Zhang, J. Wu, and L. Song, "Cellular UAV-to-device communications: Trajectory design and mode selection by multi-agent deep reinforcement learning," *IEEE Transactions on Communications*, vol. 68, no. 7, pp. 4175–4189, 2020.
[50] U. Challita, W. Saad, and C. Bettstetter, "Interference management for cellular-connected UAVs: A deep reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2125–2140, 2019.
[51] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.
[52] V. Sadhu, C. Sun, A. Karimian, R. Tron, and D. Pompili, "AerialDeepSearch: Distributed Multi-Agent Deep Reinforcement Learning for Search Missions," in *Proceedings of the IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 2020, pp. 165–173.



Omar Sami Oubbati is an Associate Professor at University Gustave Eiffel in the region of Paris, France. He is a member of the Gaspard Monge Computer Science laboratory (LIGM CNRS UMR 8049). He received his degree of Engineer (2010), M.Sc. in Computer Engineering (2011), M.Sc. degree (2014), and a PhD in Computer Science (2018), all from University of Laghouat, Algeria. From Oct. 2016 to Oct. 2017, he was a Visiting PhD Student with the Laboratory of Computer Science, University of Avignon, France. He spent 6 years as an Assistant Professor at the Electronics department, University of Laghouat and a Research Assistant in the Computer Science and Mathematics Lab (LIM) at the same university. His main research interests are in Flying and Vehicular ad hoc networks, Energy harvesting and Mobile Edge Computing, Energy efficiency and Internet of Things (IoT). He is the recipient of the 2019 Best Survey Paper for Vehicular Communications (Elsevier). He has actively served as a reviewer for flagship IEEE Transactions journals and conferences, and participated as a Technical Program Committee Member for a variety of international conferences, such as IEEE ICC, IEEE CCNC, IEEE ICCCN, IEEE WCNC, and IEEE GlobeCom. He serves on the editorial board of Vehicular Communications Journal of Elsevier and Communications Networks Journal of Frontiersin. He has also served as guest editor for a number of international journals. He is a member of the IEEE and IEEE Communications Society.



Abderrahmane Lakas received his BS degree in Computer Systems from the National Institute of Informatics, Algiers, Algeria (1989), and the MS (1990) and PhD (1996) degrees in Computer Systems from the University of Paris 6, France. He is currently a Professor at the Computer and Network Engineering Department, College of Information Technology, UAE University. Prior to joining UAE University, he had many years of industrial experience serving in various technical positions in telecommunication companies such as Netrake (Plano, Texas, 2002), Nortel Networks (Ottawa, 2000) and Newbridge Networks (Ottawa, 1998). His research interests include Network Design and Performance, Wireless Communications and Mobile Computing, Network Security, Quality of Service, Vehicular ad hoc Networks, Intelligent Transportation Systems, Unmanned Aerial Vehicles, Intelligent Autonomous Systems, Internet of Things. He is the founder and the head of the CAST (Connected Intelligent Autonomous Systems) Lab at the college of IT, UAE University. He served as a reviewer for many high impact journals and a member of the technical program committee of many international conferences. He serves as associate editor of IEEE Access, and served as member of the editorial board of Journal of Communications, and Journal of Computer Systems, Networks, and Communications. He is a member of the IEEE since 2003 and a senior member of the IEEE since 2020.



Mohsen Guizani received the B.S. (with distinction) and M.S. degrees in electrical engineering, the M.S. and Ph.D. degrees in computer engineering from Syracuse University, Syracuse, NY, USA, in 1984, 1986, 1987, and 1990, respectively. He is currently an Associate Provost for Faculty Affairs and Institutional Advancement at the Machine Learning Department, Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI), United Arab Emirates. He was a Professor at the Computer Science and Engineering Department in Qatar University, Qatar. Previously, he served in different academic and administrative positions at the University of Idaho, Western Michigan University, University of West Florida, University of Missouri, Kansas City, University of Colorado-Boulder, and Syracuse University. Dr. Guizani was selected as the Best Teaching Assistant for two consecutive years at Syracuse University. Mohsen was a professor within the Department of Electrical and Computer Engineering at the University of Idaho, where he served as the department chair. Moreover, he served as the Vice President of Graduate Studies at Qatar University, Associate Dean at Kuwait University, Chair of Computer Science Department at Western Michigan University; Chair of Computer Science Department at the University of West Florida; and Director of graduate studies at the University of Missouri-Columbia. He also taught and held administration positions at Syracuse University and the University of Colorado-Boulder. His research interests include wireless communications and mobile computing, computer networks, mobile cloud computing, security, and smart grid. He is currently the Editor-in-Chief of the IEEE Network Magazine and an Advisory Board Editor of the IEEE Internet of Things Journal. He serves on the editorial boards of several international technical journals and the Founder and Editor-in-Chief of Wireless Communications and Mobile Computing journal (Wiley). He is the author of nine books and more than 600 publications in refereed journals and conferences. He guest edited a number of special issues in IEEE journals and magazines. He also served as a member, Chair, and General Chair of a number of international conferences. Throughout his career, he received three teaching awards and four research awards. He is the recipient of the 2017 IEEE Communications Society Wireless Technical Committee (WTC) Recognition Award, the 2018 AdHoc Technical Committee Recognition Award for his contribution to outstanding research in wireless communications and Ad-Hoc Sensor networks and the 2019 IEEE Communications and Information Security Technical Recognition (CISTC) Award for outstanding contributions to the technological advancement of security. He was the Chair of the IEEE Communications Society Wireless Technical Committee and the Chair of the TAOS Technical Committee. He served as the IEEE Computer Society Distinguished Speaker and is currently the IEEE ComSoc Distinguished Lecturer. He is a Fellow of IEEE and a Senior Member of ACM.