Utilizing Deep Reinforcement Learning to Control UAV Movement for Environmental Monitoring

Thu Nga Nguyen[1], Trong Binh Nguyen[1], Trinh Van Chien[2], and Tien Hoa Nguyen[1,*]
[1] School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, Vietnam
[2] School of Information and Communication Technology, Hanoi University of Science and Technology, Vietnam
Email: nga.nguyenthu1@hust.edu.vn (T.N.N.), binh.nt182905@sis.hust.edu.vn (T.B.N.), chien.trinhvan@hust.edu.vn (T.V.C.), hoa.nguyentien@hust.edu.vn (T.H.N.)

***Abstract*—Unmanned aerial vehicles (UAVs) are increasingly used in various applications, including infrastructure inspection, traffic monitoring, remote sensing, mapping, and rescue. However, many applications have required UAVs to function autonomously, without human intervention to improve system performance. In this study, we propose a new approach to environmental monitoring using a group of UAVs equipped with sensors under the support of reinforcement learning. Regarding the communication system model, we assume that UAVs can cooperate with each other to learn and share information about the environment, and then relocate to an optimal position while managing connectivity and coverage. After that, we exploit reinforcement learning with a deep deterministic policy gradient (DDPG) algorithm to optimize environmental monitoring with the proposed algorithm. Specifically, the proposed algorithm aims to simulate an environmental monitoring system using UAVs with basic parameters. We further apply the proposed algorithm to evaluate network performance under different parameter settings. Numerical results validate the effectiveness of the proposed learning-based framework in monitoring and sensing data.**

***Index Terms*—Connectivity maintenance, coverage maximization, deep reinforcement learning, Unmanned aerial vehicles (UAVs)**

I. Introduction

Unmanned aerial vehicles (UAVs) are mobile machines that are utilized in a wide range of networked fields, including traffic monitoring, electrical system testing, and package delivery [1−3]. While UAV usage is not limited to industry and academia, it demonstrates the potential to provide a visual line of sight (VLoS), subject to certain constraints imposed by applicable regulations (e.g., flight zone or specific location) [4]. Although UAVs are primarily used in VLoS scenarios, there are many applications that may be characterized by Beyond-VLoS (BVLoS) environments as examining a large area [5−7]. Therefore, consensus among stakeholders is necessary to expand the commercial range of UAVs, especially in areas with limited visibility, such as urban areas (central cities with high buildings and large obstacles), residential areas in borders, high mountains, and islands. As cutting-edge technology trends continue to emerge, UAVs are being integrated into wireless mobile networks, fifth-generation (5G) systems, and beyond, making UAV management an essential aspect of mobile communication development [8]. New network models, such as edge computing, cloud computing, and cellular networks, can help UAVs handle high-speed flight control applications, while hardware vendors allow the integration of different microprocessor architectures into UAVs [9]. This enables UAVs to handle real-time applications and optimize radio resources for orbital control.

UAVs play a crucial role in deploying radio networks for real-world applications that meet diverse communication system requirements. Among these requirements, coverage and connectivity are considered the most critical factors [10]. Coverage refers to the ability to reliably monitor areas or targets of interest using sensors, while connectivity involves transmitting sensor data from sensors to a central processing station [11]. In many applications, it is essential to ensure both coverage and connectivity because radio networks are responsible for continuously monitoring and analyzing targets or areas [12, 13]. Moreover, mobile networks are dynamic, and a non-coverage area can cause network links to change, leading to collisions during network planning and packet transmission. Traditional algorithms to solve this problem often have high computational complexity and may not be practical for fast-fading channels [14]. Therefore, there is a need for low-cost algorithm designs for resource allocation in UAV-aided networks.

Reinforcement learning provides a mathematical framework for developing strategies or methods that map states to actions, with the goal of maximizing the cumulative reward function [15]. It has been widely applied in various fields, such as manufacturing and automation, financial policy optimization, and robotic

Manuscript received March 22, 2023; revised May 26, 2023; accepted May 30, 2023.

[*] Corresponding author

doi: 10.18178/ijeetc.12.5.317-325

control systems. With the development of deep learning techniques, reinforcement learning has evolved into Deep Reinforcement Learning (DRL), which uses deep machine learning neural networks (DNNs) in the policy formation process [16, 17]. The offline-learning structure and DNN network can predict and update online when combined with DRL, making DRL capable of handling complex problems with multidimensional data sets in the action space (even allowing the action space to be a continuous domain) [18]. These new features have made significant contributions to recent breakthroughs in smart telecommunications, where DRL has been applied to enable radio resource planning for real-time applications.

Many research papers have proposed applying deep reinforcement learning techniques to environmental monitoring. Reference [19] focused on the challenge of drone navigation in an environment with numerous obstacles, utilizing sensor data. The authors adopt the Proximal Policy Optimization (PPO) deep reinforcement learning algorithm for a single drone to reach the goal location. In [20], the authors presented a cooperative multi-UAV data collection system that works in concert to minimize the overall energy consumption of both UAVs and sensors. They proposed a deep deterministic policy gradient (DDPG)-based approach for power control and obstacle-aware navigation. Additionally, they provide a multi-UAV scheduling framework to create an activity plan for each UAV. Similarly, in [21], the aim was to maximize the total throughput of UAV-to-vehicle communications. In [22], the authors proposed a MARL algorithm that can be applied to a team of UAVs, enabling them to cooperatively learn to provide full coverage of an unknown field of interest while minimizing overlapping sections among their fields of view. In [23], the authors applied the DRL technique to the classification of detected anomalies for intelligent video surveillance applications; in particular, they used the Deep Q Learning method with deep CNN. Their validation showed promising results when applying DRL. While the research papers mentioned above offer deep reinforcement learning techniques for either a single UAV or multiple UAVs, they do not pay attention to the connectivity between UAVs.

In this paper, we investigate a novel deployment model where UAVs are utilized to sense data. In summary, our main contributions are listed as follows:

- We consider an area where the coverage is defined by a set of UAVs. In particular, the UAVs from a hub will fly and distribute in the network to sense and gather information. Collaboration between UAVs is allowed to enhance connectivity under practical conditions including sensing range and limited energy of each UAV.
- We formulate an optimization problem that maximizes the sensing range and maintains the connectivity of the network. We propose to exploit DDPG to solve this problem in polynomial time.
- Numerical results demonstrate the benefits of our proposed method in improving the sensing range and connectivity performance. In particular, the reward function is significantly improved during epochs.

II. Network Model and Markov Decision Process

A. System Model

We consider an environmental monitoring system as shown in Fig. 1. Specifically, the system consists of N UAVs that monitor an area of interest. Each UAV is equipped with a sensing function and a communication function. Equipped with sensing functions, the UAVs can collect data about the phenomenon of interest, such as gas/radiation leakage, radioactivity substance, and toxic pollutants. In addition, with a proper communication protocol, each UAV can exchange its collected data with the other UAVs in the system. The UAVs gather desired data based on the sensing functions for different purposes. The dynamic network can qualify and allocate radio resources from the collected sensing data and measurement metrics.

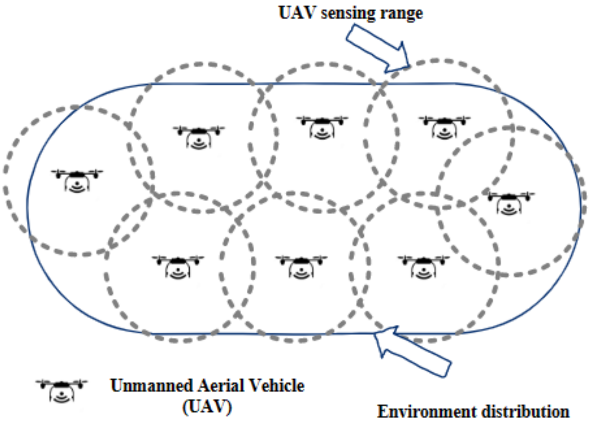

Fig. 1. The considered environment monitoring system model using multiple UAVs.

In general, various measurement metrics are combined with different ways to collect sensing data adapting to the UAV positions. In this paper, we adopt a widely-used method applied for geological and environmental sciences [24] to model the distribution of sensing data in the coverage area with the presence of UAVs. We denote $\delta(\mathbf{p}) \in \mathbb{R}$ the sensing sample defined by the UAV at position $\mathbf{p}$. Here, the position $\mathbf{p}$ is determined by a corresponding tuple in the Cartesian coordinate system, i.e., $\mathbf{p} = (x, y, z)$. Mathematically, $\delta(\mathbf{p})$ can be modeled as follows:

$$\delta(\mathbf{p}) = \boldsymbol{\beta}^T \mathbf{F}(\mathbf{p}), \tag{1}$$

where $\boldsymbol{\beta} = [\beta_1, \beta_2, \cdots, \beta_m]^T$ are the constants and $\mathbf{F}(\mathbf{p}) = [f_1(\mathbf{p}), f_2(\mathbf{p}), \cdots, f_m(\mathbf{p})]^T$ are spatial basis functions with T being the transpose operator. The subscript m denotes the number of basis functions. The kth element of $\mathbf{F}(\mathbf{p})$ is denoted by $f_k(\mathbf{p})$, $\forall k$, represents a Gaussian radial basis function that can be mathematically expressed as:

$$f_k(\mathbf{p}) = \exp\left(\frac{-\|\mathbf{p} - \mathbf{q}_k\|^2}{2\sigma_k^2} \right), \tag{2}$$

where $\mathbf{q}_k$ and σ_k^2 are the center position and variance of the basis function $f_k(\mathbf{p})$, respectively. In Eq. (2), the operator $\|\quad\|$ represents the Euclidean norm.

It is assumed that within the system under consideration, there is a special UAV, called the sink node, equipped with higher computing and energy capabilities than the remaining UAVs. The sink node periodically determines optimal movement directions and speeds for all the UAVs based on gathered information about desired positions in the network. It can also forward the collected sensing data of all UAVs to a central station for other analysis and decision-making activities. The sink node, operating as a central controller, aims to control the movement directions and speeds of all UAVs in the system, maximizing sensing coverage based on the collected data. Additionally, the network can minimize energy consumption while maintaining connectivity among the UAVs.

B. Markov Decision Process (MDP) Formulation

As previously mentioned, the sink node periodically determines the optimal movement directions and speeds for all the UAVs. To account for the variation in propagation channels over time and frequency, we can divide the time domain into intervals with a fixed duration of τ seconds, referred to as the control period, during which the channels are static. The starting time instant of a control period is referred to as a time step, and the control action is performed at each time step.

1) State Space

Let $\mathbf{p}_i$ and ϕ_i denote the position and sensing sample of the *i*th UAV at the current time step. Again, the position $\mathbf{p}_i$ has the corresponding coordinate of (x_i, y_i, z). For the sake of simplicity, this paper assumes that all the UAVs fly at the same altitude. Then, the state space of the system, denoted by $\mathcal{S}$, is defined as:

$$\mathcal{S} = \{(\mathbf{p}_1, \phi_1), \ldots, (\mathbf{p}_N, \phi_N)\}, \tag{3}$$

which comprises the positions and sensing samples of all the *N* UAVs.

2) Action Space

Let $\mathcal{A}$ denote the action space of the system. For given a certain state $s \in \mathcal{S}$, a control action $a \in \mathcal{A}$ is performed to determine the movement speed and direction of the *N* UAVs in the next control period. Thus, $\mathcal{A}$ can be defined as:

$$\mathcal{A} = \{(v_1, \alpha_1), \ldots, (v_N, \alpha_N)\}, \tag{4}$$

where $\alpha_i \in [0, 2\pi]$, $\forall i = 1, 2, \cdots, N$ is the movement direction of the UAV, and $v_i \geq 0$ presents the speed of the *i*th UAV. We emphasize that if $v_i = 0$ the *i*th UAV does not move in the next period. Alternatively speaking, this UAV hovers at the current location. Otherwise, the UAV *i* moves to the next position with the speed of v_i.

3) Reward Function

In this paper, to improve the coverage area and energy efficiency of the system, the objectives of the monitoring system comprise of:

- Maximizing the information in the sensing data, which is obtained by the UAVs.
- Maximizing the sensing coverage and minimizing the energy consumption of the UAVs.
- Maintaining connectivity among the UAVs.

Based on the above objectives, the reward function is designed as explained as follows:

Energy consumption: When the control action *a* is performed at the current time step with the system state of *s*, let $e_i(s,a)$ denote the total movement energy usage of the UAV *i* during the control period τ. In this work, we assume that each UAV consumes e_0 Joules to travel 1~[*m*] [25]. As such, one can define the energy consumption of the UAV *i*th as follows:

$$e_i(s,a) = \tau e_0 v_i, \tag{5}$$

and the energy consumption of the system is:

$$\Xi = \sum_{i=1}^{N} e_i(s,a), \tag{6}$$

which is measured over all the *N* UAVs in the system.

Sensing data and sensing coverage: It is assumed that all the UAVs are equipped with sensors with the same sampling frequency, which is denoted by *f*. Then, the number of sensing samples that the UAV *i* collects during the period τ is $M=f\tau$. We denote:

$$\phi_i(s,a) = \sum_{k=1}^{M} \phi_{i,k}, \tag{7}$$

as the sum of UAV *i*'s sensing samples during period τ where $\phi_{i,k}$ is determined according to Eq. (1). To maximize the total interest value, the UAVs should move to positions with high-interest values. The total interest value achieved by the system is $\sum_{i=1}^{N} \phi_i(s,a)$.

To maximize the sensing coverage of the system, the overlap of sensing coverage among the UAVs should be minimized. For this, we denoted r_c and r_s as the communication and sensing radius of all UAVs, respectively. We define $Y_{\text{int}}(s, a)$ as a metric for qualifying the degree of coverage and the interest value achieved the system given a pair of (*s*, *a*). Then, Y_{int} is defined as follows:

$$Y_{\text{int}} = \sum_{i=1}^{N} \phi_i(s,a) + \sum_{i=1}^{N}\sum_{j=1}^{N} \max(d_{i,j} - d_{\text{th}}, 0), \tag{8}$$

where $d_{i,j}$ is the distance between two UAVs *i* and *j*, d_{th} is the distance threshold between two adjacent UAVs. The existing works [26, 27] showed that the hexagonal pattern can maximize sensing coverage while avoiding coverage holes. To achieve goals, we set the distance threshold d_{th} to the distance between two adjacent nodes in a hexagonal pattern, i.e. $d_{\text{th}} = \sqrt{3} r_s$. From Eq. (2), if the UAVs are located at the highest interest positions and the distance between any pair of UAVs is higher than d_{th}, then Y_{int} is maximized.

Connectivity maintenance: We denote c_i as a connectivity coefficient that becomes 1 if the UAV

ith has a path to the sink node and becomes 0 otherwise. Note that the path can be a single-hop path or a multi-hop path. Given UAVs's positions $\mathbf{p}_i$, i =1, 2, ⋯, N the Dijkstras shortest path algorithm [28] can be used to find a path from ith UAV to the sink node. Let $\Psi_c = \sum_{i=1}^{N} c_i$ denote the metric for qualifying the network connectivity when the action a is performed at the state s. We define an immediate reward function as follows:

$$r(s,a) = \lambda_1 \Upsilon_{\text{int}} + \lambda_2 \psi_c - \lambda_3 \Xi, \qquad (9)$$

where λ_1, λ_2 and λ_3 are the weights associated with Y_{int}, ψ_c and Ξ, respectively. As such, the reward $r(s,a)$ is determined based on the weighted sum of the interest value, movement energy usage, degree of coverage maximization, and network connectivity maintenance.

Algorithm 1: DDPG Algorithms

```
Initialize random critic network $Q(s,a|\theta^Q)$ and actor-network with weights $\theta^Q$ and $\theta^\mu$.
Initialize target network $Q'$ and $\mu'$ with weights $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\mu'} \leftarrow \theta^\mu$
Initialize buffer $R$
for episode = 1, $M$ do
    Initialize a random process $N_t$ for action exploration.
    Receive initial observation state $s_1$
    for $t$=1, $T$ do
        Select action $a_t = \mu(s_t|\theta^\mu) + N_t$ according to the current policy and exploration noise.
        Execute action $a_t$ observe reward $r_t$ and new state $s_{t+1}$
        Store transition $(s_t, a_t, r_t, s_{t+1})$ in $R$
        Sample a random mini-batch of $N$ transitions $(s_t, a_t, r_t, s_{t+1})$ from $R$
        Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'}$
        Update critic by minimizing the loss:
            $L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$
        Update actor policy using sampled policy gradient:
            $\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_i \nabla_a Q(s,a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s_i}$
        Update target network:
            $\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$, $\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$
    end for
end for
```

Movement control problem: The controller is implemented in the sink node. At every time step, the system controller observes the system state s. Then, it decides an action a that determines the movement speed v_i and direction α_i for every i in the next control period of τ seconds. At the end of the next control period, the system controller can calculate the immediate reward $r(s,a)$ as a feedback signal. The main design objective is to find a movement control policy that decides a based on s to maximize of the expected reward over a long run, i.e., E[r]. Generally, designing a closed-form movement control policy to maximize E[r] is challenging because the area of interest is unknown and thus the state evolution of the system is complex. In this study, we apply a model-free DRL to deal with the above challenges. Through the interactions between the DRL agent and the environment, the agent learns the optimal control policy from the historical data, including system states, control actions, and the resulting immediate rewards.

III. DDPG Algorithm for Environment Monitoring

A. Algorithm Introduction

In this section, we utilize the DDPG algorithm to allocate an action strategy to the UAVs. Firstly, we provide a brief introduction to DDPG, followed by defining the DDPG states, actions, and rewards for the agent. It is worth noting that DDPG is an extension of the deep Q network (DQN) algorithm introduced by Mnih *et al.* [29], which was the first approach to combine deep learning and reinforcement learning but was limited to low-dimensional action space sets. DDPG, on the other hand, is a deep reinforcement learning algorithm capable of dealing with multidimensional action spaces, seeking to find the optimal action strategy for agents that maximizes the reward for completing a given task [30]. Unlike classical deep learning methods such as Q-learning, the DDPG algorithm can solve continuous spatial sets, which is a major obstacle.

DDPG is based on the actor-critic (Policy-Evaluation) algorithm. It's basically a method that combines gradient policy and function values. The policy function μ is called the Actor, while the value function Q is called the Critic. The agent output is essentially an action selected from a continuous action space, with the current state of the environment $a = \mu(s|\theta^\mu)$. For the Critic network, its output $Q = (s,a|\theta^\mu)$ is a signal of the error form: Time difference (TD) to evaluate the actions of the agent knowing the current state of the environment. A schematic diagram of the agent evaluation architecture is shown in Fig. 2.

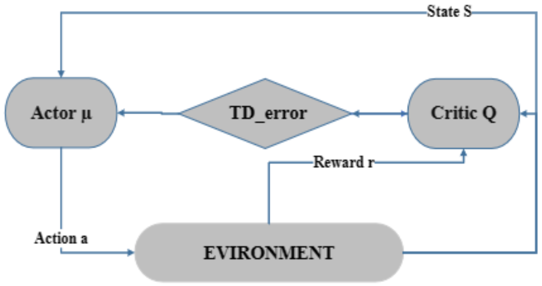

Fig. 2. Actor-Critic network structure.

There are also some practical tricks used to enhance performance. The trade-off between discovery and mining is done using the ε algorithm, where a random action a_t is chosen with probability ε, a correct action $a_t = \mu(s_t \mid \theta^{\mu})$ is selected for the current policy with probability $1-\varepsilon$. Furthermore, an experiential playback buffer b, of size B, is used during the training phase to break the temporal correlations.

Each interaction with the environment is stored as tuples of the form (s_t, a_t, r_t, s_{t+1}), which are the current state, the action to take, and the reward for performing action a in state s_t and the next state, respectively (Algorithm 1 (Line 9)).

During the learning phase, a set of data is randomly extracted from the buffer and used (Algorithm 1, line 10). Additionally, target networks are exploited to prevent algorithmic divergence caused by direct updates of the network weights with gradients obtained from the TD error signal. The DDPG algorithm is applied to the system model with the agent, which consists of 20 UAVs with one original UAV performing the task of determining the movement direction and speed of other UAVs. The action set, state set, and reward function are defined in the sub-section 'system model'.

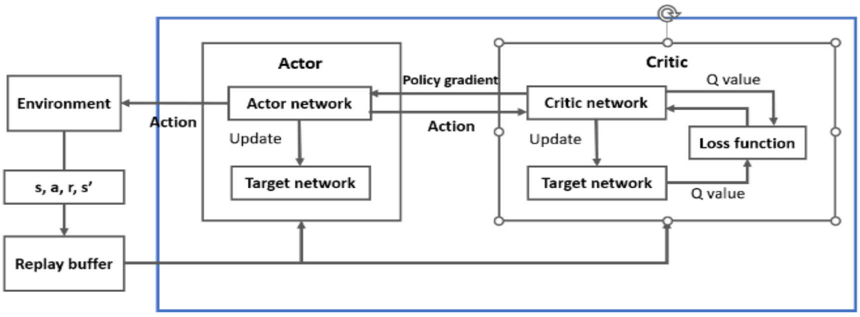

Fig. 3. DDPG algorithm framework.

B. Framework

The framework of the proposed method is illustrated by Fig. 3, which includes a critic network with parameters θ^{Q} and an actor network with parameters θ^{μ}, an actor target network with parameters $\theta^{Q'}$, a critic target network with parameters $\theta^{\mu'}$. The algorithm uses the neural network to approximate an action under the obtained policy and to approximate the Q value of a state-action pair according to the policy.

In particular, the actor network takes state s as input and output an action a, and the critic network takes state s and action a as input and output the Q value under the policy.

An experience replay D is created to allow the DRL agent of the DDPG-based algorithm to learn from its interactions with the environment. At each time step t, the agent excutes an action at under the state s_t, then it gets the reward t_t and the next state s_{t+1}. Therefore, the transition tuple (s_t, a_t, r_t, s_{t+1}) is obtained and stored in the experience replay buffer D.

The DRL agent goes through two stages of learning, called the policy evaluate stage and the policy update stage. These two stages combine with each other to find the optimal policy, which can return the optimal action according to the current state. To describe this process, we consider the kth transition tuple (s_k, a_k, r_k, s_{k+1}) is sampled from buffer D. In the policy evaluate stage, the critic network takes the state s_k and the action a_k as input, the output is the Q value denote $q(s_k, a_k)$ to evaluate the policy. The target actor network takes the state to ouput the action a_{k+1}. The target critic network takes the state s_{k+1} and a_{k+1} to get the Q value $q(s_{k+1}, a_{k+1})$. The loss function is calculated (Algorithm 1 line 12, 13), then using gradient descent method, we can minimize the loss function. The weight parameters of the critical network are updated to achieve a better Q value.

In the policy update stage, we fixed the value of Q obtained from the critic network and performed gradient ascent method to update the weight parameters of the actor network which is also the policy network. Thus, we get the policy to return the optimal action.

The parameters of target actor network and target critic network are updated using soft update method [30].

C. Model Training

The contruction of the critic network and the actor network is illustrated in Fig. 4. The critic network and the target critic network have the same structure which includes one input layer, one output layer and two hidden layers. The input dimension of the critic actor is the combination of the state space and the action space, whereas the output is one dimension. The input dimension of the actor-network is the state space, and the output dimension is the action space. The target actor network has the same structure as the actor-network.

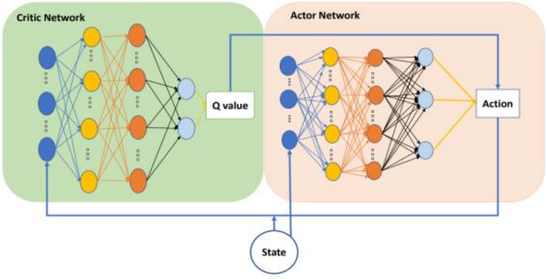

Fig. 4. Critic network and actor-network

The activation function applied in the layers of the critic and actor-network is the rectified linear unit (ReLU). The optimizer used here is the Adam optimizer.

The training process of critic network: A random mini-batch of N transitions are sampled from the experience replay buffer D. Denote the target Q value of the ith transition tuple (s_i, a_i, r_i, s_{i+1}) as y_i written as:

$$y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1} \mid \theta^{\mu'}) \mid \theta^{Q'}, \qquad (10)$$

where μ' is the policy that outputs the action a_{i+1} under the state s_{i+1} through the target actor-network $\theta^{\mu'}$. $\theta^{Q'}$ is target critic network which outputs the Q value according to the action a_{i+1}, state s_{i+1} pair. Besides $\gamma \in (0, 1]$ is the discount factor.

Minimizing the loss function between the y_i value and the output Q value from the critic network written as:

$$L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i \mid \theta^Q))^2, \qquad (11)$$

We can update the weight parameters of the critic network via gradient descent:

$$w_q = w_q - \alpha \nabla_{w_q} L\left(w_q\right), \qquad (12)$$

where α is the learning rate. The training process of the actor-network: The policy obtained from the algorithm should be updated to return the action value in the direction which increases Q values. The actor-network outputs the action value and the critic network output the Q value. Using gradient ascent, the weight parameters of the actor-network can be updated as:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_i \nabla_a Q(s, a \mid \theta^Q)|_{s=s_i, a=\mu_{(s_i)}} \nabla_{\theta^\mu} \mu(s \mid \theta^\mu)|_{s_i}, \qquad (13)$$

$$w_\mu = w_\mu - \beta \nabla_{\theta^\mu} J, \qquad (14)$$

where β is the learning rate.

IV. Simulation and Result

A. Simulation Design

We consider a network comprising a set of UAVs, with a sensing radius r_s and communication radius r_c set to 80 m and 160 m, respectively. The UAVs are placed in the environment with values of interest distributed according to the basic Gaussian function. In more detail, we consider a UAV system operating in a two-dimensional environment that is mapped into a Cartesian system with (x, y) coordinates. Our considered model corresponds to all UAVs of the same altitude. The covered system area is $1000\text{m} \times 1000\text{m}$. All the network parameter settings are given in Table I. Numerical results are implemented in the Python programming language by utilizing a personal computer with the configuration specified in Table II.

Table I: System model Parameters

Parameters	Value
Communication radius	160 m
Sensing radius	80 m
Monitoring area	1000 m × 1000 m
Numbers of UAVs	15, 20
Energy consumption coefficient	8 J/m

Table II: Configuration for Used Simulation

Component	Specification
Processor	Intel(R) Core(TM) i7-9750HF CPU @ 2.60GHz
GPU	NVIDIA GeForce GTX 1050
Driver version	527.37
RAM	8.00 GB

Agents in the system use a DDPG network structure as shown in Table III. The Policy Network has 521 nodes with FC1-2 and Output of 2, respectively. The Evaluation Network also has a value of 512 nodes with FC1-2 and Output of 1, respectively. The basic parameters are shown in the following Table IV.

Table III: Training Parameters

Parameters	Value
Training episodes	500
Beta	0.002
Gamma	0.99
Batch size	64
Noise	0.1
Optimizer	Adam

Table IV: Network Parameters

Network	Layer	Size	Activation
Actor	FC1	512	
	FC2	512	
	Output	2	Relu
Critic	FC1	512	
	FC2	512	
	Output	1	

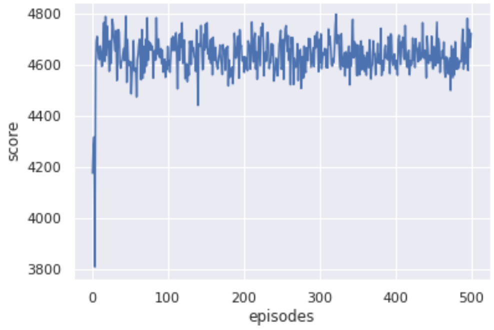

Fig. 5. Reward function with N=15.

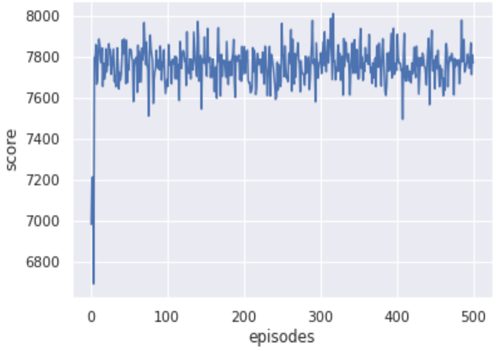

Fig. 6. Reward function with N = 20.

B. System Performance

This section presents the simulation results of the DDPG algorithm for environmental monitoring using unmanned aerial vehicles. The goal is to evaluate the effectiveness of the algorithm in improving system performance. Fig. 5 and Fig. 6 depict the value of the reward function obtained for the number of training sets of 500 in two cases where the number of UAVs is 15 and 20, respectively.

The results demonstrate that the algorithm converges around the set of 60 for both cases, indicating that the algorithm can achieve good system performance with a limited number of training sets. Moreover, with 20 UAVs in the network, there is wider coverage and a larger reward compared to the case with 15 UAVs. The reward function improves as the number of iterations increases, reaching a 20% improvement at the converged point compared to the initial setting value. Overall, the results indicate that the DDPG algorithm is a promising approach for environmental monitoring through UAVs and can significantly improve system performance. By providing a comprehensive evaluation of the algorithm's effectiveness, this study contributes to advancing the state-of-the-art in the field of UAV-based environmental monitoring.

Fig. 7 illustrates the convergence of the algorithms as a function of the number of UAVs. The results show that the convergent reward value increases with an increase in the number of UAVs, indicating that a larger number of UAVs leads to a higher system performance. For example, when there are only 20 UAVs in the network, the reward function is approximately 100. However, with 20 UAVs covering the network area, the reward function significantly increases to over 6000. These outcomes demonstrate that the DDPQ algorithm is highly effective in improving system performance, particularly in large-scale networks, and outperforms other conventional algorithms.

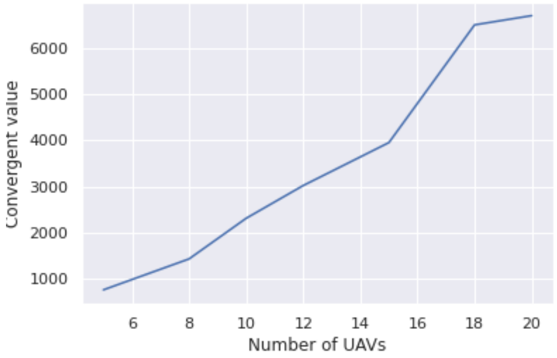

Fig. 7. Converged values for the different number of UAVs.

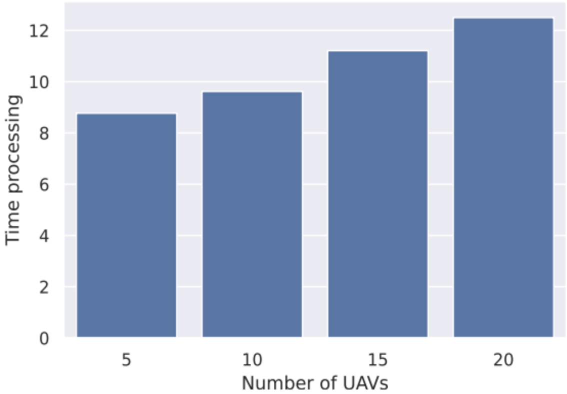

Fig. 8. Time processing for number of UAVs.

Fig. 8 depicts the processing time of the algorithms in relation to the number of UAVs. The results demonstrate that as the number of UAVs increases, it takes more time for the algorithms to reach the convergent reward value. While the DDPQ algorithm yields a higher system performance, it is accompanied by a longer processing time. Therefore, a trade-off between system performance and processing time must be considered when selecting the algorithm for practical applications. Fine-tuning the algorithm parameters to optimize computational complexity may be necessary to strike a balance between system performance and processing time. Overall, the results suggest that the DDPQ algorithm is a promising approach for improving system performance in large-scale UAV networks, but careful consideration of its computational complexity is required for practical implementation.

V. Conclusion

Based on our study, we have shown that unmanned aerial vehicles (UAVs) with a deep reinforcement learning algorithm can be used to optimize the monitoring of an environment. We have formulated an optimization problem to maximize the sensing range and maintain connectivity between the UAVs. Our proposed approach allows for collaboration between UAVs to enhance connectivity under experimental conditions, including the sensing range and limited energy of each UAV. We applied the DDPG algorithm to solve the optimization problem, which found the operational policy of the UAVs that minimized their traveling energy consumption. The simulation results demonstrated that our proposed approach outperforms existing methods in terms of sensing range and connectivity. Specifically, we achieved a sensing range of a covered system area is 1000m×1000m, with the connectivity of 15-20 UAVs. The reward function improves as the number of iterations increases, reaching a 20% improvement at the converged point compared to the initial setting value. We have demonstrated the ability to control many UAVs collaboratively, which can improve the accuracy and efficiency of data collection. Scenarios where UAVs with different heights or the different joint transmission policies between UAVs at the signal processing levels should be potential directions to extend our framework for future work.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

T. N. Nguyen proposed the idea and wrote the manuscript, T. V. Chien derived the mathematical framework and proofread the manuscript, and T. B. Nguyen oversaw data curation and software. T. H. Nguyen conducted the numerical results and proofread the paper.

Funding

This work was funded by the Hanoi University of Science and Technology (HUST) under grant number T2021-PC-015.

REFERENCES

[1] M. Khosravi and H. Pishro-Nik, "Unmanned aerial vehicles for package delivery and network coverage," in *Proc. of 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–5.

[2] G. Sun, J. Li, A. Wang *et al.*, "Secure and energy-efficient UAV relay communications exploiting collaborative beamforming," *IEEE Trans. on Communications*, vol. 70, no. 8, pp. 5401–5416, 2022.

[3] Z. Zuo, C. Liu, Q. L. Han, and J. Song, "Unmanned aerial vehicles: Control methods and future challenges," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 4, pp. 1–14, 2022.

[4] S. Ouahouah, M. Bagaa, J. Prados-Garzon, and T. Taleb, "Deep-reinforcement-learning-based collision avoidance in UAV environment," *IEEE Internet of Things Journal*, vol. 9, no. 6, pp. 4015–4030, 2022.

[5] R. J. L. Hartley, I. L. Henderson, and C. L. Jackson, "BVLOS unmanned aircraft operations in forest environments," *Drones*, vo. 6, no. 7, #167, 2022.

[6] K. H. Terkildsen, U. P. Schultz, and K. Jensen, "Safely flying BVLOS in the EU with an unreliable UAS," in *Proc. of 2021 Int. Conf. on Unmanned Aircraft Systems (ICUAS)*, 2021, pp. 591–601.

[7] I. L. Henderson and A. Shelley, "Examining unmanned aircraft user compliance with civil aviation rules: The case of New Zealand," *Transport Policy*, vol. 133, pp. 176-185, Mar. 2023.

[8] H. Yang, J. Zhao, J. Nie, *et al.*," UAV-assisted 5G/6G networks: Joint scheduling and resource allocation based on asynchronous reinforcement learning," in *Proc. of IEEE INFOCOM 2021-IEEE Conf. on Computer Communications Workshops (INFOCOM WKSHPS)*, 2021, pp. 1–6.

[9] P. Smyczynski, L. Starzec, and G. Granosik, "Autonomous drone control system for object tracking: flexible system design with implementation example," in *Proc. of 2017 22nd Int. Conf. on Methods and Models in Automation and Robotics (MMAR)*, 2017, pp. 734–738.

[10] I. Jawhar, N. Mohamed, and J. Al-Jaroodi, "UAV-based data communication in wireless sensor networks: Models and strategies," in *Proc. of 2015 Int. Conf. on Unmanned Aircraft Systems (ICUAS)*, 2015, pp. 687–694.

[11] D. Popescu, C. Dragana, F. Stoican, L. Ichim, and G. Stamatescu, "A collaborative UAV-WSN network for monitoring large areas," *Sensors*, vol. 18, no. 12, #4202, 2018.

[12] J. R. Antunes, L. Brisolara, and P. R. Ferreira, "UAVs as data collectors in the WSNs: Investigating the effects of back-and-forth and spiral coverage paths in the network lifetime," in *Proc. of 2020 X Brazilian Symposium on Computing Systems Engineering (SBESC*), 2020, pp. 1–8.

[13] M. Mozaffari, W. Saad, M. Bennis, *et al.*, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2334–2360, 2019.

[14] N. Tekin and V. C. Gungor, "Lifetime analysis of error control schemes on wireless sensor networks in industrial environments," in *Proc. of 2019 27th Signal Processing and Communications Applications Conference (SIU)*, 2019, pp. 1–4.

[15] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[16] G. Gupta and R. Katarya, "A study of deep reinforcement learning based recommender systems," in *Proc. of 2021 2nd Int. Conf. on Secure Cyber Computing and Communications (ICSCCC)*, 2021, pp. 218–220.

[17] H. Li, T. Wei, A. Ren, *et al.*, "Deep reinforcement learning: Framework, applications, and embedded implementations: Invited paper," in *Proc. of 2017 IEEE/ACM Int. Conf. on Computer-Aided Design (ICCAD)*, 2017, pp. 847–854.

[18] H. van Hasselt and M. A. Wiering, "Reinforcement learning in continuous action spaces," in *Proc. of 2007 IEEE Int. Symposium on Approximate Dynamic Programming and Reinforcement Learning*, 2007, pp. 272–279.

[19] V. J. Hodge, R. Hawkins, and R. Alexander, "Deep reinforcement learning for drone navigation using sensor data," *Neural Computing and Applications*, vol. 33, pp. 2015–2033, 2021.

[20] S. S. Khodaparast, X. Lu, P. Wang, and U. T. Nguyen, "Deep reinforcement learning based energy efficient multi-UAV data collection for IoT networks," *IEEE Open Journal of Vehicular Technology*, vol. 2, pp. 249–260, 2021.

[21] M. Zhu, X. Y. Liu, and X. Wang, "Deep reinforcement learning for unmanned aerial vehicle-assisted vehicular networks," *arXiv*, arXiv:1906.05015, 2019.

[22] H. X. Pham, H. M. La, D. Feil-Seifer, and A. Nefian, "Cooperative and distributed reinforcement learning of drones for field coverage, *arXiv*, arXiv:1803.07250, 2018.

[23] R. F. Mansour, J. Escorcia-Gutierrez, M. Gamarra, *et al.*, "Intelligent video anomaly detection and classification using faster RCNN with deep reinforcement learning model," *Image and Vision Computing*, vol. 112, #104229. 2021.

[24] N. Cressie, *Statistics for Spatial Data*, John Wiley & Sons, 2015.

[25] M. Rahimi, H. Shah, G. S. Sukhatme, *et al.*, "Studying the feasibility of energy harvesting in a mobile sensor network," in *Proc. of 2003 IEEE Int. Conf. on Robotics and Automation*, 2003 pp. 19–24.

[26] D. V. Le, H. Oh, and S. Yoon, "Virfid: A virtual force (VF)-based interest-driven moving phenomenon monitoring scheme using multiple mobile sensor nodes," *Ad Hoc Networks*, vol. 27, pp. 112–132, 2015.

[27] S. Yoon, O. Soysal, M. Demirbas, and C. Qiao, "Coordinated locomotion and monitoring using autonomous mobile sensor nodes," *IEEE Trans. on Parallel and Distributed Systems*, vol. 22, no. 10, pp. 1742–1756, 2011.

[28] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "33.3: Finding the convex hull," in *Introduction to Algorithms*, 1990, pp. 955–956.

[29] V. Mnih, K. Kavukcuoglu, D. Silver, *et al*., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[30] T. P. Lillicrap, J. J. Hunt, A. Pritzel *et al*., "Continuous control with deep reinforcement learning," *arXiv*, arXiv:1509.02971, 2015.

Nguyen Thu Nga received the MSc and PhD degrees in Electronics and Telecommunications Engineering from HUST in 2005 and 2017, respectively. Since 2005 she has worked at School of Electrical and Electronic Engineering, HUST as a lecturer and researcher. Her main areas of research interest are 5G/6G mobile communications as channel modeling, especially stochastic channel model MIMO-OFDMA systems, and channel coding for 5G/6G.

Nguyen Trong Binh is currently a fourth-year student studying an advanced program in electronics and communication engineering at Hanoi University of Science and Technology. He spent a year working in the Department of Signal Processing Research Laboratory. His research interests are in signal processing for B5G/6G systems, as well as artificial intelligence applications in big data processing.

Trinh Van Chien (Member, IEEE) received the B.S. degree in electronics and telecommunications from the Hanoi University of Science and Technology (HUST), Hanoi, Vietnam, in 2012, the M.S. degree in electrical and computer engineering from Sungkyunkwan University (SKKU), Seoul, South Korea, in 2014, and the Ph.D. degree in communication systems from Linköping University (LiU), Linköping, Sweden, in 2020. He was a Research Associate with the University of Luxembourg, Esch-sur-Alzette, Luxembourg. He is currently with the School of Information and Communication Technology (SoICT), HUST. His interest lies in convex optimization problems and machine learning applications for wireless communications and image & video processing Dr. Chien received the Award of Scientific Excellence in the first year of the 5G wireless project funded by European Union Horizon 2020. He was an IEEE Wireless Communications Letters Exemplary Reviewer in 2016, 2017, and 2021.

Nguyen Tien Hoa graduated with a Dipl.-Ing. in Electronics and Communication Engineering from Hanover University. He has worked in the R&D department of image processing and in the development of SDR-based drivers at Bosch in Germany. He spent three years experimenting with MIMOon's R&D team, developing embedded signal processing and radio modules for LTE-A/4G. After that, he worked as a senior expert at Viettel IC Design Center (VIC) and VinSmart, developing advanced solutions for aggregating, splitting, and steering traffic at the PDCP layer and above, to provide robust integration between heterogeneous link types and QoS/QoE guarantees in 5G systems. Currently, he is a lecturer at the School of Electrical and Electronic Engineering at Hanoi University of Science and Technology. His research interests include resource allocation in B5G&6G, massive MIMO, and vehicular communication systems.