Deep Reinforcement Learning Based Data Collection in IoT Networks

Seyed Saeed Khodaparast, Xiao Lu, *Member, IEEE*, Ping Wang, *Fellow, IEEE*,
Uyen Trang Nguyen, *Member, IEEE*

2022 IEEE Wireless Communications and Networking Conference (WCNC) | 978-1-6654-4266-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/WCNC51071.2022.9771616

***Abstract*—Unmanned aerial vehicles (UAVs) are an emerging technology that can be effectively utilized to perform data collection tasks in the Internet of Things (IoT) networks. However, both the UAV and the sensors in these networks are energy-limited devices, necessitating an energy-efficient data collection procedure to ensure network lifetime. In this paper, we consider a UAV-assisted network, where a UAV flies to the ground sensors according to a predetermined schedule and controls the sensor's transmit power when hovering above the sensor. Our goal is to minimize the total energy consumption of the UAV and the sensors, which is needed to accomplish the data collection mission. We formulate this problem into two sub-problems of UAV navigation and sensor power control and model each part as a finite-horizon Markov Decision Process (MDP). We deploy the deep deterministic policy gradient (DDPG) method to generate the best trajectory for the UAV in an obstacle-constrained environment and to control the sensor's transmit power during data collection. Our simulations show that the UAV can find a safe and energy-efficient path for each trip. In addition, continuous sensor power control achieves better performance against the fixed-power and fixed-rate approaches in terms of the total energy consumption during data collection.**

I. INTRODUCTION

The recent advancements in the field of Internet-of-Things (IoT) have made wireless sensor networks (WSNs) to be widely deployed to monitor the surrounding environment, e.g., temperature and air pollution [1]. These IoT sensors are usually energy-limited devices with small transmit power whose sensed data needs to be collected and sent to a control center [2]. In that respect, as a mobile data collector, an Unmanned Aerial Vehicle (UAV) can fly close to the sensors and collect their information. Compared to ground data collection schemes, the UAV is more flexible as it can easily adjust its path to reach a sensor by avoiding traffic and terrestrial obstacles [3]. In addition, due to high altitude, the UAV has a higher chance of establishing a Line-of-Sight (LoS) link to ground sensors, resulting in better link quality.

However, the UAV has limited onboard energy similar to the sensors, which poses a crucial challenge. After dispatching from the origin, the UAV needs to plan its trajectory towards the associated sensors while avoiding collisions with obstacles. On the other hand, when the UAV arrives at the sensor's location and hovers above it to start the data collection, the communication link may experience different channel gains depending on the multi-path propagation environment. Hence, the sensor's transmit power needs to be controlled efficiently to prolong its lifetime and provide reliable communication.

The UAV data collection problem has attracted increasing research attention over the past few years. Reference [4] aims to minimize the overall energy consumption of a single UAV and a set of IoT sensors by jointly optimizing the UAV's stop positions, the subset of sensors that transfer data at each stop, and the UAV's trajectory under the constraints of the individual energy availability of the sensors. However, the authors do not consider the impact of small-scale fading on the sensor transmit power. To present a more accurate and realistic model of the communication link between the UAV and the sensors, references [5], [6] adopt a Rician fading channel model. The authors in [5] optimize three-dimensional trajectory jointly with communication scheduling to maximize the minimum transmission rate of the sensors. In [6], the authors optimize the UAV trajectory and allocation of resources to maximize the total number of served IoT devices where the collected data has deadlines.

The above-reviewed optimization approaches require complete system information in advance and can only be applied in a pre-scheduled manner. Deep reinforcement learning (DRL) allows us to find a solution by letting the agent, in our case, the UAV, interact with the environment without knowing the model. DRL-based approaches have been explored in a number of UAV data collection scenarios. In [7], the authors develop a double deep Q-network (DDQN)-based trajectory planning mechanism to maximize the throughput over the whole data collection mission subject to the maximum flight time and navigation constraints, while only considering discrete flying actions for the UAV. Reference [8] proposes an Age-of-Information (AoI)-based trajectory planning algorithm for fresh data collection. The authors assume that the UAV can successfully collect the data as long as the device is in its coverage range without accurately modeling the communication channel. Reference [9] addresses obstacle-aware data collection by first obtaining the shortest trajectory based on a DRL method. Then, it determines the best schedule for visiting the sensors based on Q-learning. The proposed

The authors Seyed Saeed Khodaparast, Ping Wang, and Uyen Trang Nguyen are with the Department of Electrical Engineering and Computer Science, York University, Toronto, ON, M3J 1P3, and Xiao Lu is with Ericsson Canada, Ottawa (e-mail: skhodap@yorku.ca, superselmer@gmail.com, pingw@yorku.ca, utn@cse.yorku.ca).

approach does not consider the power consumption of the sensors.

In this paper, we propose an energy-efficient data collection approach by presenting two frameworks to address the UAV trajectory planning and sensor power control. The main contributions of our work are summarized as follows:

1) Firstly, we propose an obstacle-aware navigation framework by deploying the deep deterministic policy gradient (DDPG) algorithm [10] to help the UAV plan its trajectory towards any given destination sensor with minimum energy consumption.
2) Secondly, for the case when the UAV arrives at the sensor's location and hovers above it to collect data, we propose a DDPG-based continuous sensor power control method by considering the multi-path propagation environment and the energy consumption of the hovering UAV and the sensor.
3) Our simulation results show that the UAV learns to fly safely towards the sensors without any prior knowledge about the obstacles' positions. In addition, the power control model can successfully control the trade-off between the UAV's hovering power and the sensor's transmit power, resulting in better performance than the fixed-power and fixed-rate approaches.

II. System Model

We consider a data collection scenario with N sensors located on the region and a UAV to collect the data gathered by the sensors. The position of the UAV and ground sensor n in the 3D space is characterized by (x_U, y_U, z_U) and $(x_n, y_n, 0)$, respectively, where $x_U(x_n)$ and $y_U(y_n)$ denote the coordinates of the UAV (sensor n) on the ground and z_U denotes the UAV's altitude. The UAV starts from the origin (e.g., charging station), move towards the sensors according to the schedule, and go back to the origin when the data collection is finished. We assume that when the UAV arrives at the sensor's location, it hovers above the sensor and activates it by sending a query signal. During each communication round, the UAV can acquire the channel gain by exchanging signals and, based on that, controls the transmit power of the sensor. We adopt a discrete-time system in which time is divided into slots of T_{nav} and T_{com} length for the navigation and sensor power control tasks, respectively.

A. Environment and Navigation Model

We consider a virtual environment with dense obstacles to match the complex urban areas. Our goal is to train a UAV to fly from any starting location to any arbitrary destination while avoiding collision with the obstacles. We use spherical coordinate system to characterize the UAV's dynamic at time t by

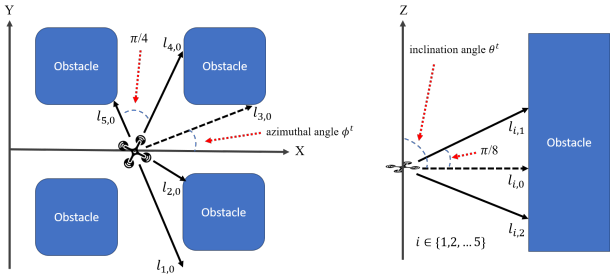

Fig. 1. A range finder structure with $m_1 = 5$ and $m_2 = 2$. $l_{1,0} \sim l_{5,0}$ denotes distances returned by the main range finders. $l_{i,1}$ and $l_{i,2}$ denote distances outside the horizontal plane given in the i-th horizontal direction.

$$\begin{aligned} x_U^{t+1} &= x_U^t + v^t\, T_{nav}\, sin(\theta^t)\, cos(\phi^t), \\ y_U^{t+1} &= y_U^t + v^t\, T_{nav}\, sin(\theta^t)\, sin(\phi^t), \\ z_U^{t+1} &= z_U^t + v^t\, T_{nav}\, cos(\theta^t), \end{aligned} \tag{1}$$

where v^t, θ^t, and ϕ^t are the UAV's speed, inclination, and azimuthal angles, respectively. We assume that the UAV knows its current location and destination by using GPS signals. To illustrate the UAV's perception of the surrounding environment, we assume that the UAV is equipped with range finder structure [11] characterized by (m_1, m_2), where m_1 is the number of range finders in the $x-y$ plane and m_2 is the number of additional range finders outside the horizontal plane that can identify the distance from the UAV to obstacles in multiple directions, as denoted in Fig. 1.

B. UAV Energy Consumption Model

The energy consumption of the UAV includes two main components, namely communication-related energy and propulsion energy. The communication-related energy is used in communication circuits of the UAV to send, receive and process the signals, which is negligible comparing to the propulsion energy [12]. We assume the communication-related power is a constant denoted by P_c. The propulsion energy is needed to keep the UAV flying and hovering, which can be expressed as a function of its speed [13], i.e.,

$$P_{nav}(v) = P_0\left(1 + \frac{3v^2}{U_{tip}^2}\right) + P_i\left(\sqrt{1 + \frac{v^4}{4v_0^2}} - \frac{v^2}{2v_0^2}\right)^{\frac{1}{2}} + \frac{1}{2} d_0 \rho s A v^3, \tag{2}$$

where P_0 and P_i are constant parameters representing the blade profile power and induced power in hovering status, U_{tip} denotes the tip speed of the rotor blade, v_0 is the mean rotor-induced velocity during hovering. Moreover, the parameter d_0, s, ρ, and A represent the fuselage drag ratio, rotor solidity, air density, and rotor disc area, respectively.

To obtain the power consumption during hovering, we let $v = 0$ in (2), which gives us the hovering power $P_h = P_0 + P_i$.

Therefore, the UAV's total power consumption in the data collection status is expressed as

$$P_{dc} = P_h + P_c. \tag{3}$$

C. Channel Model and Data Collection Rate

During data collection, the UAV hovers above the currently associated sensor at a high altitude, which increases the chance of establishing a line-of-sight (LoS) link. Hence, for the uplink ground-to-air (G2A) channel, we adopt the Rician fading channel consisting of a deterministic LoS link and a random multipath fading component. We assume that the channel between the UAV and sensor n remains unchanged within each time slot and at time slot t can be modeled as [5]

$$h_t[n] = \sqrt{\beta_t[n]} g_t[n], \tag{4}$$

where $\beta_t[n]$ is the large-scale average channel power gain, and $g_t[n]$ is the small-scale fading coefficient. Let $d_t[n]$ denote the distance between the UAV and sensor n which is given by $d_t[n] = \sqrt{(x_U - x_n)^2 + (y_U - y_n)^2 + z_U^2}$. We can express the average channel power gain as $\beta_t[n] = \beta_0 d_t^{-\alpha}[n]$, where β_0 is the average channel power gain at a reference distance of $d_0 = 1$ m, and α is the pathloss exponent. Due to the existence of the LoS path, the small-scale coefficient follows the Rician distribution [5] with unit power (i.e., $\mathbb{E}[|g[n]|^2] = 1$) parameterized by the Rician factor χ which is affected by the surrounding environment. It is worth mentioning that in addition to Rician channel model, other suitable channel models can also be applied to model the communication environment. The proposed DRL-based power control method does not require the UAV to have a prior knowledge about the channel model.

In the considered scenario, the UAV measures the channel gain from the sensor to itself based on the exchanged pilot signals. We can write the achievable data rate as

$$K_t[n] = B \log \left(1 + \frac{P_t G |h_t[n]|^2}{\sigma_t^2}\right), \tag{5}$$

where B,P_t, G, σ_t^2 are the bandwidth of the channel, transmit power of the sensor, the product of the transmitter's and receiver's antenna power gain, and noise power on the UAV side at time t, respectively. We use ζ_t to denote the available data to be collected at the sensor at time step t, which can be expressed as

$$\zeta_t^n = \zeta_{t-1}^n - T_{com} K_{t-1}[n]. \tag{6}$$

In addition, the initial data amount to be collected at sensor n is denoted by Λ_n, i.e., $\zeta_0^n = \Lambda_n$.

III. Proposed DRL frameworks for Autonomous Data Collection

In this section, we first model each of the UAV navigation and sensor power control tasks as an MDP. Then, we present the DRL algorithm used to solve both tasks.

A. MDP Formulation

In an MDP problem, an agent interacts with the environment by performing actions in discrete time steps. Specifically, at each time step t, the agent observes state s_t, takes an action a_t, receives a reward r_t based on s_t and a_t, and then goes to the next state s_{t+1} with probability $p(s_{t+1}|s_t, a_t)$ given by the state transition probability $\mathcal{P}$. Our goal is to find an optimal policy which performs actions for the agent over a finite horizon of length T to maximize the discounted sum of rewards starting from an initial state, i.e.,

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi,\mathcal{P}}[\sum_{t=0}^{T} \gamma^t r_t], \tag{7}$$

where γ is the discount factor balancing the importance given to the immediate and future rewards. We consider the UAV as our agent to perform navigation and sensor power control in the environment. Now, we define the state, action, and reward function for each task.

1) UAV Navigation

Suppose the UAV is located at the start point (x_U^0, y_U^0, z_U^0), aiming to fly towards sensor n. The relationship between the UAV and the target sensor at time step t can be written as

$$(\psi_x^t, \psi_y^t, \psi_z^t) = (x_U^t, y_U^t, z_U^t) - (x_n, y_n, z_{target}). \tag{8}$$

where z_{target} is the desired altitude considered for hovering. We can denote the overall state of the UAV by $s_{nav}^t = (\psi_x^t, \psi_y^t, \psi_z^t, v^t, \phi^t, l_{1,0}^t, \ldots, l_{m_1,m_2}^t)$, where $l_{1,0}^t \sim l_{m_1,m_2}^t \in [0, 100]$ show the output of the range finders. The UAV can change its speed, inclination, and azimuthal angles along the path, which can be expressed as

$$\begin{aligned} v^{t+1} &= v^t + \Delta v^t, \\ \theta^{t+1} &= \theta^t + \Delta\theta^t, \\ \phi^{t+1} &= \phi^t + \Delta\phi^t, \end{aligned} \tag{9}$$

where Δv^t, $\Delta\theta^t$, $\Delta\phi^t$ represent the change in each signal. We denote the action of the UAV by $a_{nav}^t = (\Delta v^t, \Delta\theta^t, \Delta\phi^t)$ in which all the elements are continuous variables.

The reward function must be designed to guide the UAV to reach its destination while satisfying the constraints. Our proposed reward function is composed of 4 parts: transition, energy consumption penalty, obstacle penalty, and finishing reward. The transition reward is designed as $r_{trans} = \lambda_1 \Delta d$ where λ_1 is a positive constant, and Δd shows the reduced distance to the sensor after taking the action. Δd is positive when the UAV becomes closer to the sensor, motivating the UAV to head for the sensor. We define the energy consumption penalty as $r_e = -\lambda_2 P_{nav}(v)$ where λ_2 is a positive constant, and $P(v)$ is the power consumption of the UAV defined in 2. The UAV receives this penalty to adjust its speed and trajectory so that the energy consumption along the way is minimized. To encourage the UAV to avoid the obstacles, the obstacle penalty is designed as $r_{obs} = -\lambda_3 e^{-\lambda_4 l_{min}}$ where

l_{min} is the minimum value among the range finders. If the UAVs becomes close to an obstacle in either of the range finder directions, the penalty would increase exponentially. To further encourage the UAV to move towards the sensor, it would get a large constant positive reward r_{finish} when it arrives at the destination. The overall reward function for the navigation framework is formulated as follows

$$r_{nav} = r_{trans} + r_e + r_{obs} + r_{finish}. \tag{10}$$

2) Sensor Power Control

After arriving at the target sensor, the UAV sets a proper transmit power for the sensor at each time step. Since the channel gain can be measured by the UAV, we take the channel gain into the state space to provide the UAV with better decisions. Low channel gain indicates that the sensor must transmit more power for a certain amount of data rate. Also, the amount of data that remains at the sensor must be considered since it helps the UAV achieve the goal state in which the remaining data must be zero. The state can be written as $s^t_{spc} = (g_t, \zeta_t)$ where g_t is the channel gain at time t, and ζ_t is the remaining data at the sensor. The action is denoted by $a^t_{spc} = P_t$, where P_t is the transmit power of the sensor controlled by the UAV.

For the reward function, we need to motivate the UAV to collect data, i.e., having a high data rate. In addition, we must avoid using high transmit power if the channel is not in good condition. We should note that during data collection, the UAV itself is spending power for communication and hovering as stated in (3). With all these into consideration, we design the reward function as follows:

$$r_{spc} = \lambda_5 T_{com} K_t - \lambda_6 P_{dc} - \lambda_7 P_t, \tag{11}$$

where K_t is the data rate in (5), P_{dc} is the UAV's power consumption during data collection, and P_t is the transmit power of the sensor. λ_5, λ_6 and λ_7 are the coefficients that control the tradeoff between these three components.

B. DRL Approach

We adopt the DDPG algorithm to solve both formulated MDPs. DDPG is an actor-critic DRL method, which can be applied to continuous control problems, unlike Q-learning-based algorithms that limit the agent's actions to have discrete values. The actor and critic are implemented by using neural networks, where the critic $Q(s_t, a_t|\Theta^Q)$ with parameters Θ^Q gives the actor feedback of how good its actions are, and the actor uses a deterministic policy $\pi(s_t|\Theta^\pi)$ with parameters Θ^π to generate an action.

The proposed DDPG algorithm is presented in Algorithm 1. The actor and critic networks are randomly initialized at the start of the algorithm (Line 1). We use target networks π' and Q' to improve learning stability. The target networks have the same structure as the online actor or critic networks. We initialize their weights in the same way as their online networks (Line 2). The target networks use soft updates to slowly track the online network weights (Line 14). The agent takes its actions according to the actor network and a noise process $\mathcal{N}_t$ (Line 7) to ensure the agent's exploration of the environment. Otherwise, we will not be able to try different actions since the deterministic policy outputs just a single action. DDPG uses an experience replay buffer to store the transition tuples (s_t, a_t, r_t, s_{t+1}) (Line 9), which increases the data efficiency and reduce the correlation between consecutive samples. Then, a mini-batch is randomly sampled to train the actor and critic networks (Line 10). We update the critic network weights by minimizing the loss function defined in Line 12. The sample-based version of the policy gradient is used to update the actor weights (Line 13).

Algorithm 1: The Proposed DDPG Algorithm

1: Randomly initialize critic $Q(s, a|\Theta^Q)$ and actor $\pi(s|\Theta^\pi)$ with weights Θ^Q and Θ^π
2: Initialize target network Q' and π' with weights $\Theta^{Q'} \leftarrow \Theta^Q$ and $\Theta^{\pi'} \leftarrow \Theta^\pi$
3: Initialize the replay buffer W
4: **for** episode = 1,2, ... **do**
5: Receive the initial observation state s_1
6: **for** t = 1,2, ..., T^{DDPG}_{max} **do**
7: Select action $a_t = \pi(s_t|\Theta^\pi) + \mathcal{N}_t$ according to the current policy and exploration noise
8: Execute action a_t, receive reward r_t and observe the new state s_{t+1}
9: Store transition (s_t, a_t, r_t, s_{t+1}) in W
10: Sample a random minibatch of I transitions (s_i, a_i, r_i, s_{i+1}) from W
11: Set $y_i = r(s_i, a_i) + \gamma Q'(s_{i+1}, \pi'(s_{i+1}|\Theta^{\pi'})|\Theta^{Q'})$
12: Update critic by minimizing the loss: $L(\Theta^Q) = \frac{1}{I}\sum_i \left[(Q(s_i, a_i|\Theta^Q) - y_i)^2\right]$
13: Update actor using the sampled policy gradient: $\nabla_{\Theta^\pi} J \approx \frac{1}{I}\sum_i \left[\nabla_a Q(s, a|\Theta^Q)|_{s=s_i, a=a_i} \nabla_{\Theta^\pi} \pi(s|\Theta^\pi)|_{s_i}\right]$
14: Update the target networks: $\Theta^{Q'} \leftarrow \eta\Theta^Q + (1-\eta)\Theta^{Q'}$, $\Theta^{\pi'} \leftarrow \eta\Theta^\pi + (1-\eta)\Theta^{\pi'}$
15: **end for**
16: **end for**

IV. Simulation Results

In this section, we discuss the experimental setting and the performance of the proposed frameworks. Our simulations are performed with Python 3.7 and Tensorflow 2.0. The simulation parameters regarding the UAV's propulsion power and communication channel are shown in Table I. The values of the coefficients λ_1 to λ_7 in (10) and (11) are carefully set

Parameter	Description	Simulation Value
f_c	Carrier frequency	2 GHz
B	Bandwidth	1 MHz
σ_t^2	Noise power in the given bandwidth	-100 dB
α	Pathloss exponent	2
Λ_n	Data size of each sensor	100 Mbits
U_{tip}	Tip speed of the rotor blade	200
v_0	Mean rotor induced velocity in hovering	7.2
d_0	Fuselage drag ratio	0.3
ρ	Air density	1.225
κ	Rotor solidity	0.05
A	Rotor disc area	0.79
P_0	Blade profile power in hovering	580.65
P_i	Induced power in hovering	790.67

TABLE I
PARAMTERS USED IN THE PAPER

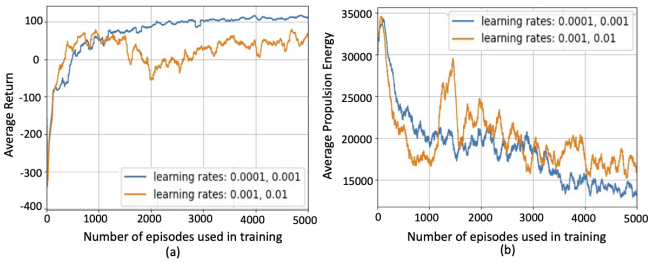

Fig. 2. Convergence of the DDPG model during the training for the navigation task: (a) the average accumulated rewards received by the UAV in the last 100 episodes (b) the average propulsion energy consumed by the UAV in the last 100 episodes.

through many trials that well balance the trade-off between different terms in the reward functions.

A. UAV Navigation

For the sake of easy presentation, the UAV's flight altitude z_U is set to 50 m, and the obstacles are randomly located in a square of size 600×600 m^2, with the same altitude as the UAV. We consider a range finder design with $m_1 = 5$ and $m_2 = 0$. The reward is instantiated as: $\lambda_1 = 0.3$, $\lambda_2 T_{nav} = 0.002$, $\lambda_3 = 50$, $\lambda_4 = 0.1$. r_{finish} is 50 at time of arrival, and 0 for the rest of the time steps.

First, we analyze the convergence of the proposed method for two cases of learning rates. We train the DDPG model for 5000 episodes. Fig. 2 (a) shows the average accumulated rewards obtained by the UAV in the last 100 episodes. Since we randomly choose the starting and finishing points with different distances between them in each episode, we average over the last 100 episodes to observe the rewards received by the agent in different scenarios. We can see that the UAV learns to obtain higher rewards at the end of the training. In Fig. 2 (b), we present the average propulsion energy consumed by the UAV in the last 100 episodes. The energy penalty defined in the reward function causes the UAV to get higher rewards by decreasing the energy consumption on its path to the destination. Although the learning rates of 10^{-3} and 10^{-2} for the actor and critic networks perform better at the beginning, they result in lower performance comparing to

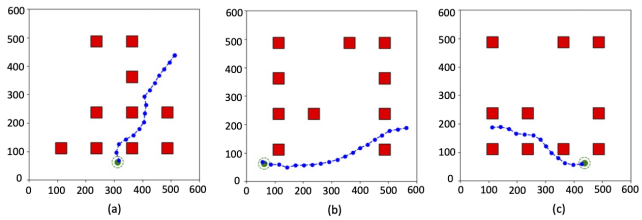

Fig. 3. UAV's trajectory in three different environment configurations.

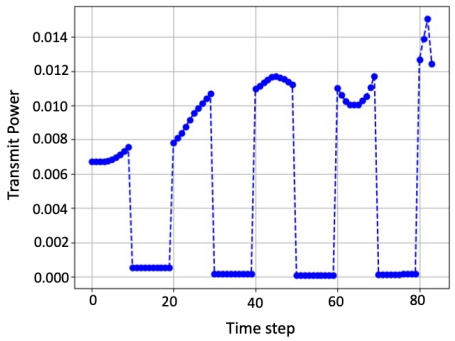

Fig. 4. The DDPG agent's output power during the considered scenario.

learning rates of 10^{-4} and 10^{-3}. High learning rates cause the gradient updates to be large, leading to a sub-optimal solution. From now on, we set the learning rate as the latter one.

In Fig. 3, we present the trajectory followed by the trained UAV for three different environments, where the green circle shows the hovering location above the sensor, the red squares show the positions of the obstacles, and the blue dots indicate the trajectory of the UAV at each time step. We change the locations of the obstacles, starting position, and the destination and observe that the trained DRL agent is flexible to different scenarios because we considered the relative position to the destination and the range finders in our state to make proper decisions in different scenarios.

We also test the model for 1000 episodes to validate the successful navigation without any collision to obstacles. The UAV reaches the target safely for 90.8% of the test episodes.

B. Sensor Power Control

Considering the reward function defined in (11), our goal is to jointly minimize the UAV and sensor's power consumption during data collection. We assume the transmit power range of the sensor is between 0 and 0.1 W. We use the following parameters for the reward function: $\lambda_5 T_{com} = 0.1$, $\lambda_6 P_{dc} = 0.1$, $\lambda_7 = 10$.

To better understand the relationship between the output power and the channel gain, we temporarily ignore the Rician distribution and assume that the channel gain in each episode can only switch between 0.1 and 1 values, where 0.1 indicates a very poor channel condition and 1 indicates a good channel condition. In Fig. 4, the sensor's output power is illustrated in each time step of an episode. In the first 10 time steps, the channel gain is set to 1, and after each 10 time steps, it switches to the other value, i.e., the channel alternates

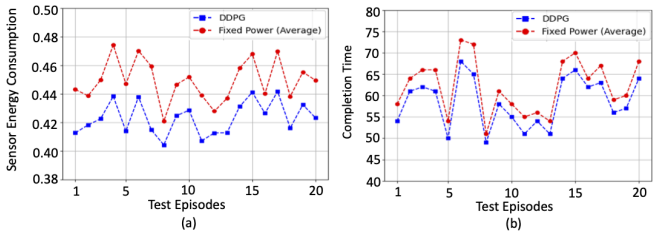

Fig. 5. Comparison between the DRL agent and the average fixed-power approach in terms of the sensor energy consumption and completion time.

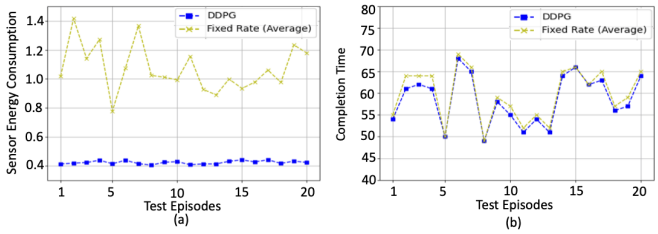

Fig. 6. Comparison between the DRL agent and the average fixed-rate approach in terms of the sensor energy consumption and completion time.

between poor and good conditions. The agent outputs a higher power as the channel link experiences better quality, whereas it outputs close-to-zero power when the channel condition is poor to avoid the waste of sensor energy. Also, as the agent approaches the end of the data collection period, it outputs a higher power to finish the task in the predefined time.

In Fig. 5, we compare the sensor power consumption and the completion time of the DRL agent against an average fixed-power approach in which the sensor outputs a fixed power equal to the average power of the DRL agent in each episode. Our goal is to see how the overall energy consumption and completion time in each episode change if the sensor transmits data with the average power of the DRL agent, regardless of the channel conditions. We perform our experiment in the environment with Rician factors χ=1. It can be seen that the DDPG agent outperforms the fixed power approach by adaptively controlling the sensor's transmit power according to the varying channel gain. The DRL agent can finish the data collection task earlier, while the sensor consumes less energy than the other method. As the completion time decreases, the UAV also spends less energy hovering. Therefore, our DDPG approach is successfully optimizing the energy consumption of the sensor and the hovering UAV.

We adopt a similar comparison between the DDPG algorithm and a fixed-rate approach, where in each episode, the fixed data transmission rate is set to the average rate of the DDPG agent. In Fig. 6, we see the fixed-rate approach consumes much higher energy, although the completion time is close to the DDPG agent. The reason is that when the link between the UAV and the sensor has a low channel gain, the fixed-rate approach needs to output a high power in order to ensure the same rate, which leads to the high energy consumption of the sensor.

V. Conclusion

In this work, we have proposed a DRL-based approach to solving the data collection problem with the aim to minimize the energy consumption of the UAV and the sensors. We have divided the original problem into two sub-problems of UAV navigation and sensor power control and solved each sub-problem by utilizing the DDPG algorithm. Our simulation results have shown that the UAV can efficiently fly to its target sensor by avoiding obstacles. In addition, by controlling the sensor's transmit power, the DDPG model performs better than the fixed-power and fixed-rate approaches. As our next step, we will consider multiple UAVs collaboratively for collecting data from multiple sensors at different locations.

References

[1] G. Ding and *et al*., "An amateur drone surveillance system based on the cognitive internet of things," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 29–35, 2018.

[2] O. Ghdiri and *et al*., "Energy-efficient multi-uav data collection for iot networks with time deadlines," *arXiv preprint arXiv:2009.06838*, 2020.

[3] C. Y. Tazibt and *et al*., "Wireless sensor network clustering for uav-based data gathering," in *2017 Wireless Days*. IEEE, 2017, pp. 245–247.

[4] M. B. Ghorbel, D. Rodríguez-Duarte, H. Ghazzai, M. J. Hossain, and H. Menouar, "Joint position and travel path optimization for energy efficient wireless data gathering using unmanned aerial vehicles," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2165–2175, 2019.

[5] C. You and R. Zhang, "3d trajectory optimization in rician fading for uav-enabled data harvesting," *IEEE Transactions on Wireless Communications*, vol. 18, no. 6, pp. 3192–3207, 2019.

[6] M. Samir, S. Sharafeddine, C. M. Assi, T. M. Nguyen, and A. Ghrayeb, "Uav trajectory planning for data collection from time-constrained iot devices," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 34–46, 2020.

[7] H. Bayerlein, M. Theile, M. Caccamo, and D. Gesbert, "Uav path planning for wireless data harvesting: A deep reinforcement learning approach," *arXiv preprint arXiv:2007.00544*, 2020.

[8] C. Zhou, H. He, P. Yang, F. Lyu, W. Wu, N. Cheng, and X. Shen, "Deep rl-based trajectory planning for aoi minimization in uav-assisted iot," in *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, 2019, pp. 1–6.

[9] O. Bouhamed, H. Ghazzai, H. Besbes, and Y. Massoud, "A uav-assisted data collection for wireless sensor networks: Autonomous navigation and scheduling," *IEEE Access*, vol. 8, pp. 110 446–110 460, 2020.

[10] 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," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[11] C. Wang, J. Wang, Y. Shen, and X. Zhang, "Autonomous navigation of uavs in large-scale complex environments: A deep reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2124–2136, 2019.

[12] C. Zhan and Y. Zeng, "Aerial–ground cost tradeoff for multi-uav-enabled data collection in wireless sensor networks," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1937–1950, 2019.

[13] 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.