IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 9, 2020, accepted October 19, 2020, date of publication October 23, 2020, date of current version November 5, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3033333

Deep Deterministic Policy Gradient With Prioritized Sampling for Power Control

SHIYANG ZHOU, YUFAN CHENG, (Member, IEEE), XIA LEI, (Member, IEEE), AND HUANHUAN DUAN

National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, Chengdu 611731, China

Corresponding author: Yufan Cheng (chengyf@uestc.edu.cn)

This work was supported in part by the National Nature Science Foundation of China under Grant U19B2014, in part by the National Key Research and Development Program of China under Grant 254, and in part by the Fund Project of the National Key Laboratory of Science and Technology on Communications under Grant 2102181402.

ABSTRACT Reinforcement learning is a technique for power control in wireless communications. However, most research has focused on the deep Q-network (DQN) scheme, which outputs the Q-value for each discrete action, and does not match the continuous power control problem. Hence, this paper provides a deep deterministic policy gradient (DDPG) scheme for power control. A power selection policy designated an actor is approximated by a convolutional neural network (CNN), and an evaluation of a policy designated a critic is approximated by a fully connected network. These deep neural networks enable fast decision-making for large-scale power control problems. Moreover, to speed up the training process, this paper proposes a prioritized sampling technique, which samples the experiences that need to be learned with a higher probability. This paper simulates the proposed algorithm in a multiple sweep interference (MSI) scenario. The simulation results show that the DDPG scheme is more likely to achieve optimal policy than the DQN scheme. In addition, the DDPG scheme with prioritized sampling (DDPG-PS) converges faster than the DDPG scheme with uniform sampling.

INDEX TERMS Power control, reinforcement learning, deep deterministic policy gradient, prioritized sampling, multiple sweep interference.

I. INTRODUCTION

In wireless communication systems, interference can degrade performance due to the broadcast nature of radio [1]. Therefore, interference mitigation has played an important role in wireless communications [2]. Power control is an effective solution for mitigating interference [3], and can be posed as an optimization problem [2], [4], [5]. These optimization techniques are suboptimal solutions for power control, while reinforcement learning-based schemes have been shown to exhibit potential for achieving the optimal policy by observing the rewards of trial-and-error interactions with the environment [6], [7].

A. MOTIVATION

Power control is often used to address the problem of interference [8]. This paper considers scenarios where multiple Linear Frequency Modulation (LFM) interferences are present, which are modeled as multiple sweep interference (MSI). MSI can seriously degrade the performance of wireless communication systems [9].

Power control [8], [10]–[13] has been studied for decades, and typical algorithms that significantly enhance performance have been proposed, such as the weighted minimum mean square error (WMMSE) algorithm [4] and the iterative algorithm based on fractional programming (FP) [5]. On the one hand, both algorithms require full channel state information (CSI), which is difficult to accurately evaluate. On the other hand, due to the dynamic characteristics of MSI, it is difficult for the WMMSE and FP algorithms to estimate MSI instantaneous parameters in a limited timeframe. Reinforcement learning is one of the most appropriate branches of machine learning to solve a complex control problem [2] that does not require full CSI and estimation of interference parameters. Therefore, the model-free reinforcement learning algorithm is a better scheme to solve

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

the problem when the channel state transition probability is unknown. Through continuous interaction with the environment, the policy is adjusted according to the feedback to optimize the performance. Hence, this paper studies reinforcement learning-based power control algorithms in MSI scenarios.

B. RELATED WORK

In the early years of power control solutions based on reinforcement learning, the Q-learning algorithm [14] was used as a practical technique to store the reward of state-action pairs in a table. In [15], without full CSI, the Q-learning power control policy traded off transmission efficiency and cost based on game theory, which improved anti-interference performance, but it learned slowly in terms of the large-scale power control problem. The decentralized Q-learning algorithm proposed in [16] optimized the transmit power of user equipment (UE) without waiting for transmission control from the base station (BS), but it was only applicable to a limited state number. In general, the Q-learning algorithm is not suitable for decision-making in continuous or large-scale states because the Q-table stores limited state-action pairs.

To this end, the deep Q-network (DQN) algorithm proposed in [17] used a neural network to map the relationship between the total discount reward and state-action pairs. Furthermore, in [18], the DQN algorithm was improved by using two identical neural networks for learning and decision-making. This approach reduced the estimation error of the Q-value by cutting off the correlation between data samples and network training. The algorithm in [18] has been widely used for the power control problem. In [19]–[21], transmit power was determined to improve transmission efficiency by the DQN scheme without being aware of the dynamic interference model, and a convolutional neural network (CNN) was used to accelerate the learning speed. Although the dimension of the state space in [19]–[21] was not large enough, power control is a continuous problem, while the decision action of the DQN algorithm is a discrete value. Thus, optimal power may not be determined. A distributively executed model-free power allocation algorithm based on the DQN scheme was developed in [22], and the approach achieved a near-optimal policy. The authors in [23] proposed a multiagent DQN-based power control method for a multiuser video transmission system, and instantaneous CSI for each link was not needed in the model-free method. In [19]–[21], discrete action was used in continuous power control, and this problem also existed in [22] and [23]. The only way to reduce the quantization error is to increase the number of output neurons in the neural network, but this increases the computational complexity. Therefore, the DQN algorithm is not the best solution for power control.

The policy gradient (PG) algorithm [24] has solved this problem by approximating the action space using a neural network. The actor-critic architecture [25] with independent policy and value networks takes advantage of value approximation and policy approximation. An actor network is used to represent the policy, and a critic network is used to approximate the action value function. The original actor-critic algorithm is difficult to converge and needs to be improved. Thus, the authors in [26] proposed a deep deterministic policy gradient (DDPG) algorithm, which is regarded as an actor-critic algorithm with deterministic policy, and is more robust than the original actor-critic algorithm. The DDPG algorithm was applied in [27] to adjust the transmit power in a frequency hopping system; it improved performance in the interference environment compared with Q-learning algorithm, but it still did not achieve an optimal policy. In [28], a DDPG network was used to dynamically adjust the resource allocation policy according to the feedback of a nonorthogonal multiple access (NOMA) system, while the learning process was not robust. The main methods used for power control are compared in Table 1.

TABLE 1. Comparison of methods for power control.

Algorithm	Optimization-based	Reinforcement learning-based		
		Q-learning	DQN	DDPG
Continuous State	Yes	No	Yes	Yes
Continuous Action	Yes	No	No	Yes
Needs CSI	Yes	No	No	No
Optimal policy	No	Yes	Yes	Yes

C. CONTRIBUTION AND PAPER ORGANIZATION

In this paper, we consider a power control problem in the MSI scenario. To maximize the reward function, the agent needs to control the transmit power on each channel of the communication system. Due to the complexity of MSI, it is difficult to evaluate the parameters of all LFM interferences. Therefore, as a model-free scheme, the DDPG scheme for power control is investigated in this paper without being aware of the interference model. Thus, we provide a DDPG scheme for power control. To solve the problem of the DQN scheme not being applicable for continuous action space decisions, we use an actor network to approximate the transmit power in this state, and the state-action pair acts as the critic network. A critic network is used to approximate the action value function in this state. The critic network is trained to minimize the mean square error of the state-action value approximation through experience. The actor network is trained to maximize the Q-value, which is the output of the critic network. Then, we study the experience sharing approach to centrally train [29], [30] the network. To further speed up the training process, we propose a prioritized sampling technique [31] based on the DDPG scheme. By sampling the experience that is more worthy of learning, the training process beconmes more effective. The contributions of this paper are summarized as follows:

1) A DDPG scheme is introduced for power control in the MSI environment, which is model-free and has been shown to be more stable than the existing DQN algorithm.

2) A centralized training scheme is proposed for the multi-task power control problem which gathers experiences from all channels in a memory pool and leads to much faster learning than distributed training techniques.
3) We propose a prioritized sampling technique for the DDPG scheme by which poorly trained experiences are sampled more frequently via nonuniform sampling. In this way, the training process is accelerated significantly.
4) Simulations show that the DDPG scheme for power control is significantly more robust than the DQN scheme. Furthermore, with prioritized sampling, the training converges much faster than in uniform sampling.

The rest of the paper is organized as follows. We present the communication system model and interference model in Section II. In Section III, we provide the DDPG scheme for power control in the MSI scenario and propose the DDPG scheme with a prioritized sampling technique. We provide simulation results in Section IV and draw the conclusions in Section V. The summary of notations used in this paper is listed in Table 2.

TABLE 2. List of Notations.

Symbol	Meaning
ε	Environment
N_c	The number of channels
Z	Summation of sensed interference and noise power vector
z_n	Summation of sensed interference and noise power on channel n
P	Transmit power vector
p_n	Transmit power on channel n
P_m	Maximum transmit power
g_p	Channel gain from transmitter to receiver
g_I	Channel gain from interferers to receiver
r_n	Reward function of user on channel n
σ^2	Variance of Gaussian white noise
C_p	Unit power cost
I_n	Interference power on channel n
$\overline{r}$	Average reward per channel
P_I	Power of a single LFM interference
$f_k^{(t)}$	Frequency channel the kth LFM interference blocks at time step t
R_k	Sweep rate
$x^{(t)}$	Current channel screen at time step t
$s^{(t)}$	State made up with $x^{(t)}$ and last $N_c - 1$ screen
$s_n^{(t)}$	Transformed state on channel n at time step t
$a_n^{(t)}$	The decided transmit power on channel n at time step t
$r_n^{(t)}$	Reward of user on channel n at time step t
γ	Discount factor of future reward
$G_n^{(t)}$	Future discounted return on channel n at time step t
$e_n^{(t)}$	Experience tuple on channel n at time step t
D	Memory pool
ρ^D	State distribution in memory pool D
w	Parameters of evaluate critic network
w'	Parameters of target critic network
u	Parameters of evaluate actor network
u'	Parameters of target actor network
η_w	Learning rate of critic network
η_u	Learning rate of actor network
τ	Soft replacement
f_i	Nonnegative Q-value of experience i
α	Priority factor
$p(i)$	Sampling probability of experience i

II. SYSTEM MODEL

A. COMMUNICATION SYSTEM MODEL

We consider a point-to-point communication scenario where a few interferers transmit LFM that interferes the communication between two users. A communication user acts as an agent that selects transmit power on each channel interacting with environment ε, and the environment ε is a complex model that can be modeled as MSI. We assume that the transmitter and receiver are stationary, and only a line-of-sight path exits in this scenario. Thus, only the large-scale fading component needs to be considered. The scenario is shown in Fig. 1.

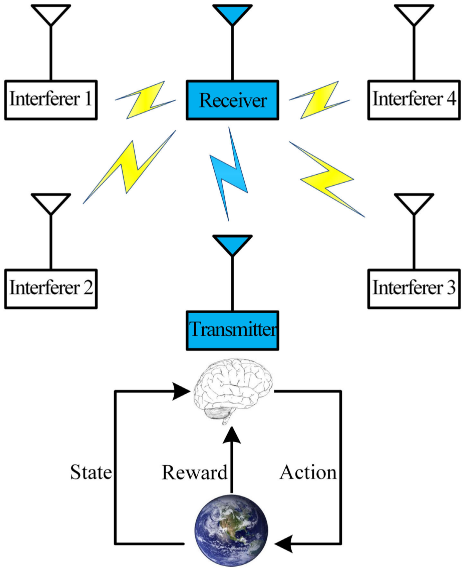


FIGURE 1. The point-to-point communication scenario with 4 LFM interferers.

We assume that the user can only sense interference and noise power on each channel, and the summation of sensed interference and the noise power vector is denoted as $Z = \{z_1, z_2, \ldots z_{N_c}\}$, where z_n is the sensed power on channel n, and N_c is the number of channels. The user specifies transmit power vector $P = \{p_1, p_2, \ldots p_{N_c}\}$ to maximize average reward performance per channel defined as (2), in which p_n is the transmit power on channel n, and $0 \leq p_n \leq P_m$, where P_m represents the maximum transmit power.

The strategy between the user and interference is formulated as a zero-sum game. The user's goal is to balance the signal-to-interference plus noise ratio (SINR) and the transmit power. The reward performance benefits from the SINR, while the cost increases with transmit power. For simplicity, we assume that all the interferers are at the same distance from the receiver. Let g_p and g_I denote the channel gains from the transmitter and interferers to the receiver, respectively. Then, the reward of channel n is defined as the utility function of user r_n, which can be formulated as (1):

$$r_n(p_n) = \frac{g_p p_n}{\sigma^2 + g_I I_n} - C_p p_n, \tag{1}$$

where σ^2 is the variance of Gaussian white noise, C_p is the unit power cost and I_n is the interference power on channel n. Then, the average reward performance per channel is formulated as (2):

$$\bar{r}(P) = \frac{1}{N_c}\sum_{n=1}^{N_c} r_n(p_n). \tag{2}$$

More specifically, the power control problem can be written as

$$\begin{aligned} &maxmize \quad \bar{r}(P) \\ &subject\ to \quad 0 \leq p_n \leq P_m, \quad n = 1, \ldots, N_c. \end{aligned} \tag{3}$$

B. INTERFERENCE CHANNEL MODEL

We formulate such an MSI model: N_K LFM interferences cyclically sweep on the entire frequency channels, each of which only blocks a single channel at a time step. The probability distributions of the sweep rate and initial frequency of these interferences are random, uniform and independent. The frequency channel of the kth LFM interference blocks at time step t is

$$f_k^{(t)} = \left(f_k^{(0)} + t \cdot R_k\right) \bmod N_c, \tag{4}$$

where $f_k^{(0)}$ and R_k are the initial frequency channel and sweep rate of the kth interference respectively.

Thus, the MSI on channel n (whose frequency is f_n) at time step t is the summation of all LFM interferences on the channel:

$$I_n^{(t)} = P_I \sum_{k=1}^{N_K} \mathbb{I}\left(f_k^{(t)}, f_n\right), \tag{5}$$

in which P_I is the power of a single LFM interference, and $\mathbb{I}(x, y)$ is an indicator function, where $\mathbb{I}(x, y) = 1$ if $x = y$ and $\mathbb{I}(x, y) = 0$ otherwise.

The time-frequency pattern of the channel with MSI is shown in Fig. 2.

III. PROPOSED ALGORITHM

A. DETERMINISTIC POLICY GRADIENT ALGORITHM FOR POWER CONTROL

The interaction process between agents and interferences is formulated as a Markov decision process (MDP). At each time step t, the agent observes the current channel screen $x^{(t)}$, which forms state $s^{(t)} = \left\{x^{(t-N_c+1)}, x^{(t-N_c+1)}, \ldots x^{(t)}\right\}$ with the last $N_c - 1$ screen, and then selects an action $a_n^{(t)} \in [0, P_m]$ based on policy $\pi\left(a_n^{(t)} \middle| s^{(t)}\right)$ on channel n for $1 \leq n \leq N_c$. The reward $r_n^{(t)}$ of action $a_n^{(t)}$ in state $s^{(t)}$ is defined as (1). We define the future discounted return at time step t as

$$G_n^{(t)} = \sum_{t'=t}^{\infty} \gamma^{t'-t} r_n^{(t')}, \tag{6}$$

where γ is the discount factor. Similarly, we define the value of taking action $a_n^{(t)}$ in state $s^{(t)}$ under policy $\pi\left(a_n^{(t)} \middle| s^{(t)}\right)$, denoted $Q(s, a)$, as the expected return:

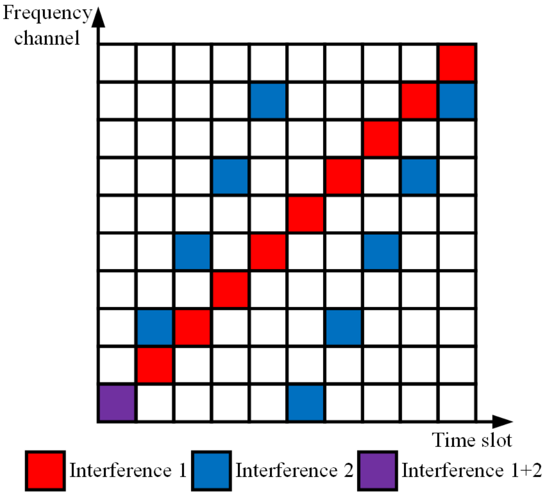


FIGURE 2. The time-frequency pattern of the channel with MSI: 2 interferences in 10 channels.

$$Q(s, a) = E_\pi\left[G \middle| s^{(t)} = s, a_n^{(t)} = a\right]. \tag{7}$$

We define the optimal action-value function as (8):

$$Q^*(s, a) = \max_\pi\left[G \middle| s^{(t)} = s, a_n^{(t)} = a, \pi\right]. \tag{8}$$

The Bellman equation is typically used to find the optimal action-value, and (8) can be rewritten as:

$$Q^*(s, a) = E_{s' \sim \varepsilon}\left[r + \gamma \max_{a'} Q^*\left(s', a'\right) | s, a\right], \tag{9}$$

where s' and a' are the next state and the action, respectively.

However, this algorithm is only applicable to models where both the state space and the action space are discrete. Quantization makes the continuous state space and action space discrete, but it introduces additional errors and increases search complexity. The actor-critic algorithm represents the continuous state space and action space by two function approximators, such as neural networks; thus, it is more suitable for the power control problem. The DPG algorithm represents policy by a neural network $a = \pi(s; u)$ as an actor network, which is a deterministic function, and represents action value by a neural network $Q(s, a; w)$ as a critic network.

We apply two identical networks, one of which is named the evaluate network, to select action and update parameters. The other network named the target network, is used to calculate the target action value. The parameters of the target network are updated with the evaluate network by soft replacement τ, which periodically updates the parameters of the target network.

The experience replay approach is used to remove correlations in the observation sequence and smooth the data distribution. We store the agent's experience tuple $e_n^{(t)} = \left(s^{(t-1)}, a_n^{(t-1)}, r_n^{(t)}, s^{(t)}\right)$ in memory pools $D_n = \left\{e_n^{(1)}, e_n^{(2)}, \ldots\right\}$ at each time step t [18] and sample the mini-batch of experience $(s, a, r, s') \sim U(D)$ randomly from the memory pool during training.

We define the objective function as the total discount reward:

$$J(u) = E_{r_t, s_t \sim \varepsilon, a_t \sim \pi}\left[G_1 \,|\pi(s; u)\right]. \tag{10}$$

Then the actor network parameters u are updated by gradient descent as (11), shown at the bottom of the page, where ρ^D is the state distribution in the memory pool.

Then, the actor network parameters u are updated as:

$$u \leftarrow u + \eta_u \nabla_u L(w), \tag{12}$$

where η_u is the learning rate of the actor network. The mean-squared error of $Q(s, a; w)$ is minimized by the loss function (13):

$$L(w) = E\left[\left(r + \gamma Q\left(s', \pi\left(s'; u'\right); w'\right) - Q(s, a; w)\right)^2\right], \tag{13}$$

where u' and w' are the parameters of the target actor network and target critic network, respectively. Then, the critic network parameters w are updated as:

$$w \leftarrow w + \eta_w \nabla_w L(w), \tag{14}$$

in which η_w is the learning rate of the critic network.

B. EXPERIENCE SHARING

The calculation rule of the reward function for each channel is the same while the mapping among the state, action and reward is different. Note that the state can be transformed so that each channel keeps the same mapping among the transformed state, action and reward. It is reasonable to gather the experience from N_c channels in one memory pool and train on the same network as long as the experience is pre-processed properly. Therefore, the training process will be accelerated if training by a central agent compared with training as N_c distributional tasks.

To achieve this goal, we change the observed state with a row transform. We denote the current observed state $s^{(t)} = \left[o_1^{(t)}, o_2^{(t)}, \ldots, o_n^{(t)}\right]^T$, where $o_n^{(t)}$ is the last N_c sensed information sequence up to time t on channel n. We define the row of transformed states starting from the current channel; then, the transformed state on channel n is $s_n^{(t)} = \left[o_n^{(t)}, o_{n+1}^{(t)}, \ldots, o_{N_c}^{(t)}, o_1^{(t)}, \ldots, o_{t,n-1}\right]^T$, as shown in Fig. 3. In this way, the experience from all channels is equivalent. Thus, the shared memory pool collects N_c times experience as before, and the experience sharing approach saves most of the storage resources and computational resources. The central DDPG algorithm is illustrated in Algorithm 1.

C. PRIORITIZED SAMPLING

There are two levels to design when using an experience replay [31] approach: which experiences to store and which experiences to replay. This paper focuses on the latter, which makes the most effective use of the stored experience for training.

Uniform sampling is not an effective sampling method; instead, prioritized sampling is a more reasonable choice. The key point of prioritized sampling is to decide the importance of each experience in the memory pool, so we sample these experiences with different probabilities. To accelerate the convergence process of the networks, poorly-trained experiences are more valuable for convergence and are given a higher probability of sampling, which improves the overall performance. For the actor network, the training goal is to maximize the Q-value, which is the output of the critic network, and for the critic network, the goal is to minimize the temporal difference (TD) error between the output and the target Q-value. For the power control problem, the action space is squeezed in 0 to P_m. Thus, the output of the actor network is designed as a sigmoid function. However, the gradient of the sigmoid function easily becomes very small and even vanishes so that the parameters of the neural network are hard to update according to the chain rule. In most cases, the training actor network is slower than the training critic network. Therefore, instead of the TD error [32], the Q-value is more relevant for prioritized sampling. The experiences with lower Q-values need to be sampled more frequently, which prevents an agent from making the wrong decision again. Considering that the Q-value is nonnegative when the optimal policy is adopted, we use a minimum function to ensure that well-trained experiences with nonnegative Q-values are sampled uniformly, as in (15).

$$f_i = \min(Q_i, 0), \tag{15}$$

Then, we formulate a nonlinear mapping between the sampling factor and sampling probability of experience i and normalize the sampling probability as (16):

$$p(i) = \frac{\exp(-\alpha f_i)}{\sum_k \exp(-\alpha f_k)}, \tag{16}$$

in which the parameter α is named the priority factor and the factor determines how much prioritization is used, and when $\alpha = 0$ prioritized sampling degenerates to uniform sampling.The DDPG scheme with prioritized sampling (DDPG-PS) for power control is illustrated in Algorithm 2.

D. NEURAL NETWORK STRUCTURE

The actor of the DDPG scheme is a CNN that consists of six layers. To better predict future channel states, CNN extracts

$$\begin{aligned} \nabla_u J(u) &\approx E_{s_t \sim \rho^U}\left[\nabla_u Q(s, a; w)\,|_{s=s_t, a=\pi(s_t; w)}\right] \\ &= E_{s_t \sim \rho^U}\left[\nabla_a Q(s, a; w)\,|_{s=s_t, a=\pi(s_t; w)} \nabla_u \pi(s; u)\,|_{s=s_t}\right] \end{aligned} \tag{11}$$

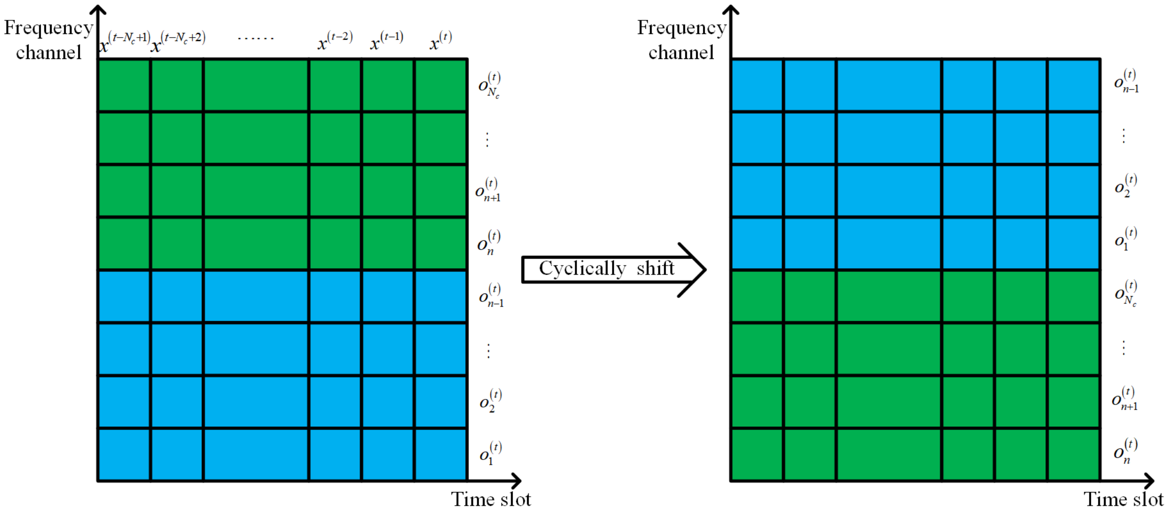


FIGURE 3. The approach of cyclically shifting the state matrix.

Algorithm 1 Centralized DDPG Scheme for Power Control

```
Input: Actor network structure, critic network structure, and MSI environment simulator
Output: Actor network with parameters $u$, critic network with parameters $w$, the policy $a$ represented by the output of actor network $\pi\ (s; u)$
Randomly initialize evaluate actor network $\pi\ (s; u)$ with weights $u$ and evaluate critic network $Q\ (s, a; w)$ with weights $w$.
Initialize target actor network $\pi\ \left(s; u'\right)$ and target critic network $Q\ \left(s, a; w'\right)$ with weights $u' \leftarrow u, w' \leftarrow w$.
Initialize the memory pool.
for $episode = 1, \dots, M$ do
    Receive initial observation state $s^{(0)}$, action $a_n^{(0)}$ and form $s_n^{(0)}$.
    for $t = 1, \dots, T$ do
        Reset a random process $N$ whose variance decays along with episode for action exploration.
        for $n = 1, \dots, N_c$ do
            Observe reward $r_n^{(t)}$ and new state $s^{(t)}$.
            Cyclically shift state $s^{(t)} = \left[o_1^{(t)}, o_2^{(t)}, \dots, o_n^{(t)}\right]^T$, the new state is $s_n^{(t)} = \left[o_n^{(t)}, o_{n+1}^{(t)}, \dots, o_{N_c}^{(t)}, o_1^{(t)}, \dots, o_{t,n-1}\right]^T$.
            Store experience tuple $e_t = \left(s_n^{(t-1)}, a_n^{(t-1)}, r_n^{(t)}, s_n^{(t)}\right)$ in memory pool $D$.
            Select action $a_n^{(t)} = \pi\ \left(s_n^{(t)}; u\right) + N_t$ according to the evaluate policy and exploration noise.
            Execute action $a_n^{(t)}$.
        end for
        Sample a random minibatch of experience tuple uniformly from memory pool $D$.
        Update evaluate critic network parameters by minimizing the loss $L\ (w)$ according to (13).
        Update evaluate actor network parameters using the sampled policy gradient as (11).
        Update the target networks: $u' \leftarrow \tau u + (1-\tau)\, u', w' \leftarrow \tau w + (1-\tau)\, w'$.
    end for
end for
```

the time-frequency pattern of MSI using a convolution kernel. The first layer is the input state matrix with $N_c \times N_c$ neurons, which consists of N_c time step information and N_c channel information. To train the network more quickly, the input state is normalized by $1/\left(\sigma^2 + I_n\right)$. The last layer is the output action with 1 neuron, and a sigmoid function is applied so that the output value ranges from 0 to 1. The output value is expanded by P_m times to ensure that the real action goes from 0 to P_m. More specifically, other parameters of the actor network are shown in Table 3.

The critic of the DDPG scheme is a fully connected network that consists of four layers. The first layer is made up of a state matrix with $N_c \times N_c$ neurons and action with 1 neuron, and it consists of $N_c^2 + 1$ in total. The last layer is the Q-value

Algorithm 2 DDPG Scheme With Prioritized Sampling for Power Control

```
Input: Actor network structure, critic network structure, and MSI environment simulator
Output: Actor network with parameters $u$, critic network with parameters $w$, the policy $a$ represented by the output of actor network $\pi\,(s;u)$
Randomly initialize evaluate actor network $\pi\,(s;u)$ with weights $u$ and evaluate critic network $Q\,(s,a;w)$ with weights $w$.
Initialize target actor network $\pi\left(s;u'\right)$ and target critic network $Q\left(s,a;w'\right)$ with weights $u' \leftarrow u$, $w' \leftarrow w$.
Initialize the memory pool.
Initialize $f_n^{(0)} = 0$, for $1 \le n \le N_c$.
for $episode = 1, \ldots, M$ do
    Receive initial observation state $s^{(0)}$, action $a_n^{(0)}$ and form $s_n^{(0)}$.
    for $t = 1, \ldots, T$ do
        Reset a random process $N$ whose variance decays along with episode for action exploration.
        for $n = 1, \ldots, N_c$ do
            Observe reward $r_n^{(t)}$ and new state $s^{(t)}$.
            Cyclically shift state $s^{(t)} = \left[o_1^{(t)}, o_2^{(t)}, \ldots, o_n^{(t)}\right]^T$, the new state is $s_n^{(t)} = \left[o_n^{(t)}, o_{n+1}^{(t)}, \ldots, o_{N_c}^{(t)}, o_1^{(t)}, \ldots, o_{t,n-1}\right]^T$.
            Store experience tuple $e_t = \left(s_n^{(t-1)}, a_n^{(t-1)}, r_n^{(t)}, s_n^{(t)}, f_n^{(t)}\right)$ in memory pool $D$ with $f_n^{(t)} = \min_{k<t} f_l^{(k)}$.
            Select action $a_n^{(t)} = \pi\left(s_n^{(t)};u\right) + N_t$ according to the evaluate policy and exploration noise.
            Execute action $a_n^{(t)}$.
        end for
        Sample a random minibatch of experience tuple by probability (16) from memory pool $D$.
        Compute the Q-value, which is output of the critic network, and update sampling factor $f$ as (15) in the memory pool.
        Update evaluate critic network parameters by minimizing the loss $L\,(w)$ according to (13).
        Update evaluate actor network parameters using the sampled policy gradient as (11).
        Update the target networks: $u' \leftarrow \tau u + (1-\tau)\,u'$, $w' \leftarrow \tau w + (1-\tau)\,w'$.
    end for
end for
```

TABLE 3. Parameters of actor network.

Layer Type	Convolution	Pooling	Convolution	Pooling	Full connection
Input Layer	Input	Conv 1	Pooling 1	Conv 2	Pooling 2
Output Layer	Conv 1	Pooling 1	Conv 2	Pooling 2	Output
Input dimension	$N_c \times N_c$	$N_c \times N_c \times 8$	$N_c/2 \times N_c/2 \times 8$	$N_c/2 \times N_c/2 \times 16$	N_c^2
Output dimension	$N_c \times N_c \times 8$	$N_c/2 \times N_c/2 \times 8$	$N_c/2 \times N_c/2 \times 16$	$N_c/4 \times N_c/4 \times 16$	1
Activation	ReLU	Max	ReLU	Max	Sigmoid
Kernel Size	3×3	/	3×3	/	/
Strider	1	/	1	/	/
Padding	/	Same	/	Same	/

TABLE 4. Parameters of critic network.

Layer Type	Full connection	Full connection	Full connection
Input Layer	Input	Hidden 1	Hidden 2
Output Layer	Hidden 1	Hidden 2	Output
Input dimension	N_c^2+1	$N_c^2/4$	$N_c^2/16$
Output dimension	$N_c^2/4$	$N_c^2/16$	1
Activation	ReLU	ReLU	Linear

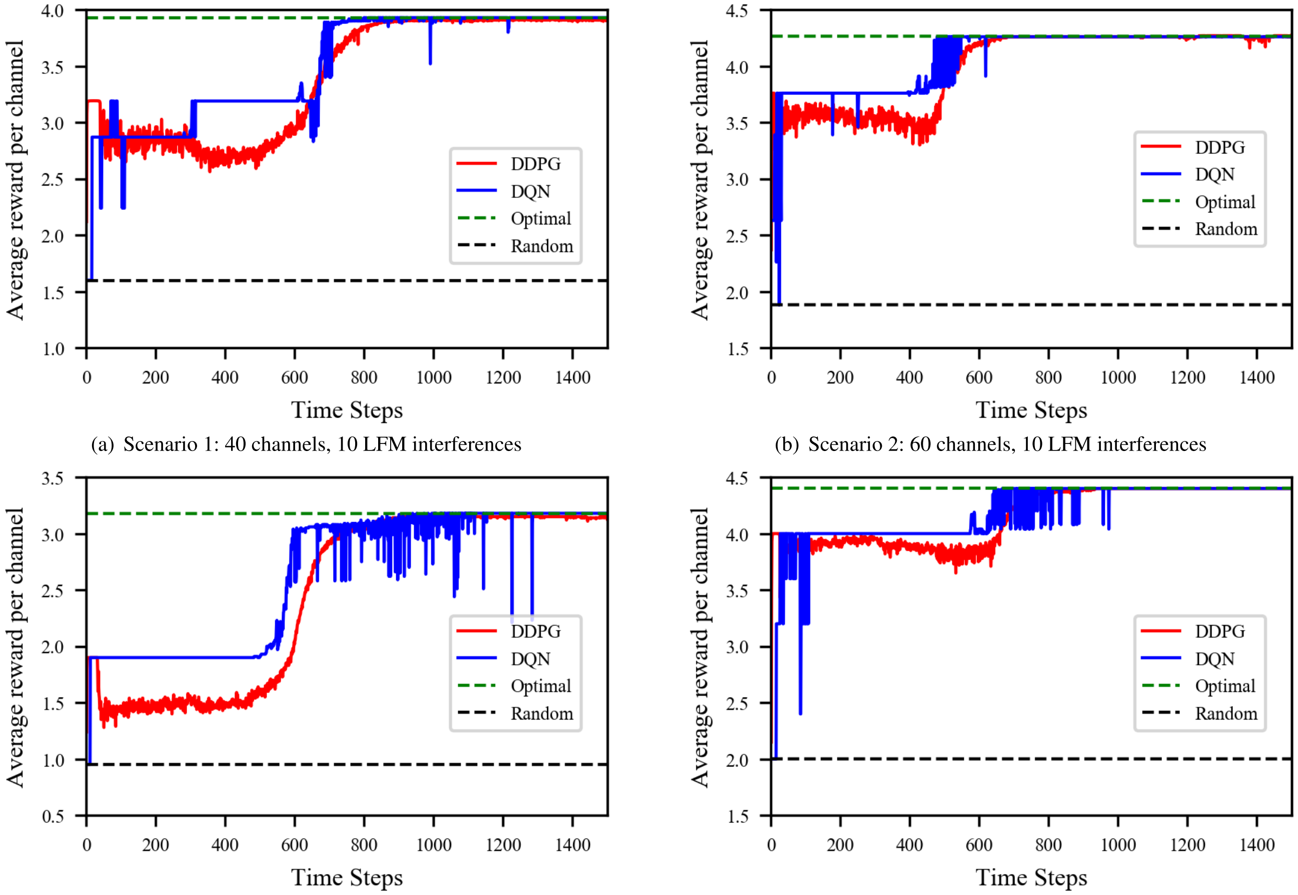
with 1 neuron, which evaluates how good the state-action is. More specifically, other parameters of the critic network are shown in Table 4.

IV. SIMULATION RESULT

In this section, our simulation results are presented to demonstrate the performance of the proposed DDPG scheme for power control. More specifically, we compare the performance of the DDPG scheme and DQN scheme in the MSI scenario. Moreover, we provide our performance comparisons to characterize the effects of prioritized sampling imposed on the achievable performance of the DDPG scheme. We simulate the schemes in four typical scenarios as illustrated in Table 5.

A. THE PERFORMANCE COMPARISON OF THE DDPG SCHEME AND DQN SCHEME

In this subsection, we compare the performance of the DDPG scheme and the DQN scheme. More specifically, the DQN structure with 2-D CNNs proposed in [21] is adopted in the simulation to be consistent with the structure of the DDPG scheme proposed in this paper. To reduce the error in simulation, we simulate for 9 times on each scheme and take the median performance according to the convergence speed,



(a) Scenario 1: 40 channels, 10 LFM interferences

(b) Scenario 2: 60 channels, 10 LFM interferences

(c) Scenario 3: 40 channels, 20 LFM interferences

(d) Scenario 4: 40 channels, 5 LFM interferences

FIGURE 4. **Average reward per channel of the DDPG scheme and the DQN scheme in different scenarios.**

TABLE 5. **Simulation scenarios.**

Scenario No.	1	2	3	4
Number of channels	40	60	40	40
The maximum transmit power	100	100	100	100
Single LFM interference power	20	20	20	20
Noise Power	1	1	1	1
Unit Power Cost	0.5	0.5	0.5	0.5
Channel gain g_P from transmitter to receiver	0.1	0.1	0.1	0.1
Channel gain g_I from interferers to receiver	0.25	0.25	0.25	0.25
Number of LFM interferences for training	[1,20]	[1,30]	[1,20]	[1,20]
Number of LFM interferences for testing	10	10	20	5

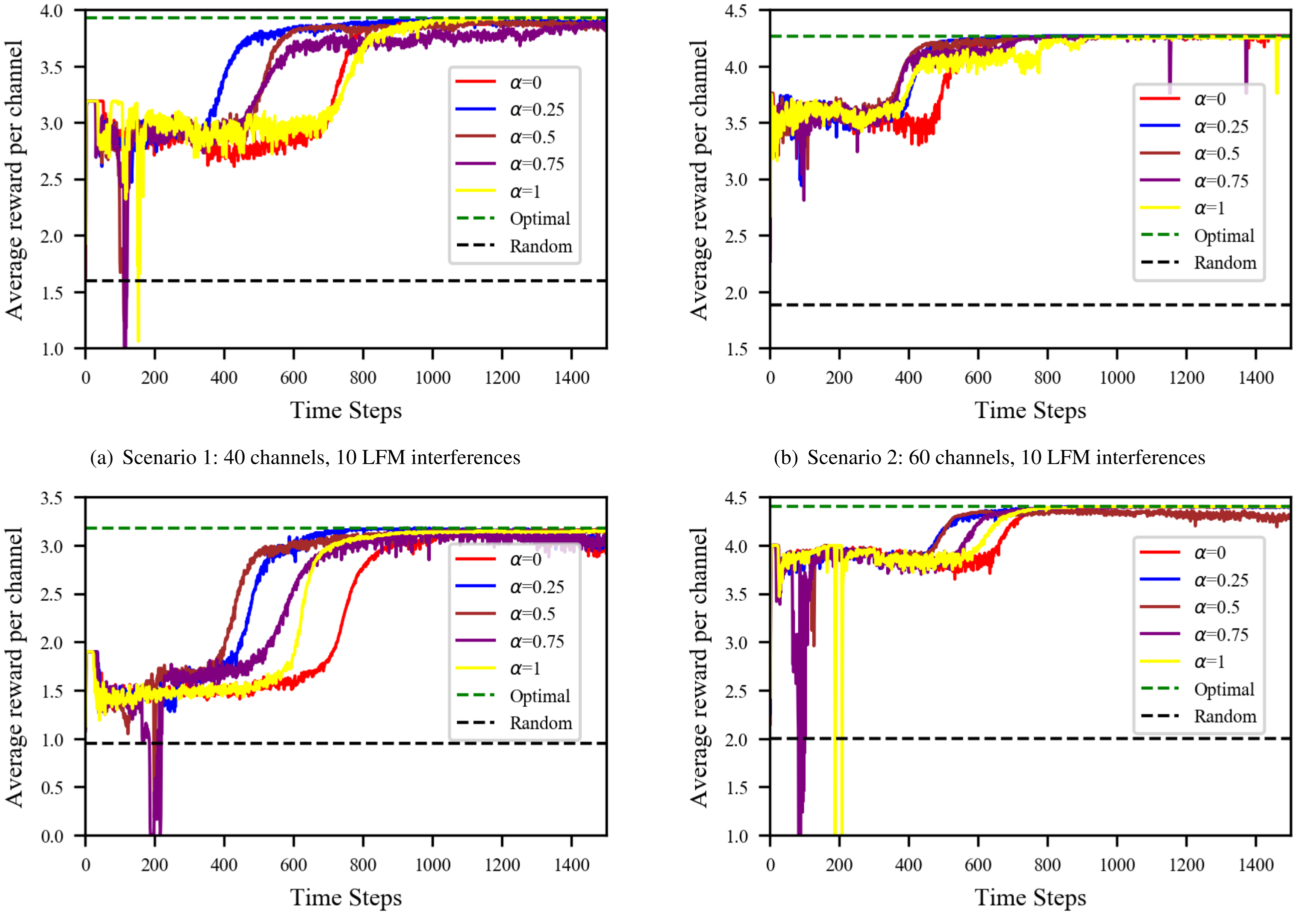
TABLE 6. **Reward performance statistics comparison between the DDPG scheme and the DQN scheme.**

Scenario No.	Scheme	Minimum reward	Maximum reward	Proportion of achieving 95% of optimal performance
1	DDPG in this paper	3.75	3.91	100%
	DQN in [21]	3.54	3.93	44.4%
2	DDPG in this paper	4.24	4.27	100%
	DQN in [21]	3.41	4.27	44.4%
3	DDPG in this paper	3.03	3.16	100%
	DQN in [21]	2.54	3.18	44.4%
4	DDPG in this paper	4.37	4.4	100%
	DQN in [21]	3.96	4.4	55.6%

as shown in Fig. 4. The final reward statistics between the two schemes are illustrated in Table 6.

For our application, the Q-network structure of the DQN scheme is the same as the actor network of the DDPG scheme except for the output dimension. More specifically, the input layer consists of N_c^2 neurons, while the output dimension depends on the quantization level. We use 11 discrete power levels. Thus, the output layer consists of 11 neurons, each of which represents the Q-value at the power level. The rectified linear unit (ReLU) activation function is applied on each layer to avoid the gradient vanishing problem of backpropagation.

Most training parameters are set to the same values in both schemes to eliminate the influence of training parameters as much as possible. For generalization, we set mini-batch size and capacity of the memory pool to 32 and 20000, the Adam optimizer [33] is used with a learning rate of 0.0001, the reward discount factor γ is set to 0.9, and the



(a) Scenario 1: 40 channels, 10 LFM interferences

(b) Scenario 2: 60 channels, 10 LFM interferences

(c) Scenario 3: 40 channels, 20 LFM interferences

(d) Scenario 4: 40 channels, 5 LFM interferences

FIGURE 5. Average reward per channel of priority factor "α=0,0.25,0.5,0.75,1" in difference scenarios.

soft replacement factor τ is 0.01 to update the target network every 100 steps. In the DQN scheme, ε-greedy algorithm is applied, the initial exploration factor is 0.1, and the exploration factor decays with the episode; the final exploration factor is 0.01. In the DDPG scheme, the initial variance of exploration is 3 and decays by 0.9999 at each time step.

Fig. 4 shows the average reward performance per channel among our proposed DDPG scheme, DQN scheme, optimal power control scheme and random power control scheme. The simulation lasts for 1500 time steps, and both schemes finally achieve optimal performance. Note that the training speed depends on the scheme and the number of channels. More specifically, the DDPG scheme learns slightly slower than the DQN scheme, because two neural networks of the DDPG scheme need to be trained in coordination, which costs a few training steps. However, since the DDPG scheme achieves the optimal policy, the performance is stable at the optimal value. In contrast, the performance of the DQN scheme is unstable, especially in scenarios 3 and 4. Moreover, it learns faster when $N_c = 60$ than when $N_c = 40$, which shows that experience sharing can speed up the training process by collecting more experience.

Table 6 shows the robustness statistics of the two schemes. In these scenarios, DDPG simulation results always achieve near optimal performance, while only approximately half of the DQN simulation results achieve that, and the other part of the DQN simulation results fell into suboptimal performance. Therefore, for the power control problem in the MSI scenario, the DDPG scheme is much more robust than the DQN scheme.

B. THE IMPROVEMENT OF PRIORITIZED SAMPLING

In this subsection, we investigate the performance of the DDPG scheme employing different priority factors. Similarly, we simulate for 9 times on each priority factor and take the median performance according to convergence speed, as shown in Fig. 5. The final reward statistics among different priority factors are illustrated in Table 7.

The learning parameters are the same as in the last subsection. Fig. 5 shows the average reward performance per channel of the DDPG scheme with different priority factors. In scenarios 1, 3 and 4 ($N_c = 40$), the two priority factors that accelerate the training process the most are 0.25 and 0.5. Among them, the DDPG scheme with the priority sampling

TABLE 7. Reward performance statistics comparison among different priority factors of the DDPG scheme.

Scenario No.	Priority factor	Minimum reward	Maximum reward	Proportion of achieving 95% optimal performance
1	0	3.75	3.91	100%
	0.25	3.78	3.93	100%
	0.5	3.78	3.93	100%
	0.75	3.74	3.92	100%
	1	3.19	3.93	77.8%
2	0	4.24	4.27	100%
	0.25	4.21	4.27	100%
	0.5	4.25	4.27	100%
	0.75	4.11	4.27	100%
	1	3.86	4.27	88.9%
3	0	3.03	3.16	100%
	0.25	2.94	3.17	100%
	0.5	3.02	3.17	100%
	0.75	0.32	3.17	88.9%
	1	0.3	3.18	77.8%
4	0	4.37	4.4	100%
	0.25	3.54	4.4	100%
	0.5	4.31	4.4	100%
	0.75	4.32	4.4	100%
	1	0.5	4.4	77.8%

technique saves up to 36.4%, 35.6%, and 25.7% time steps to achieve the optimal performance compared with the DDPG scheme with uniform sampling (when $\alpha = 0$) in scenarios 1, 3, and 4, respectively. In addition, a factor that is too high will not cause an increase in training speed; DDPG schemes with $\alpha = 1$ even cost 8.1% more time steps in scenario 1. In scenario 2, the DDPG scheme learns fastest when $\alpha = 0.5$ and $\alpha = 0.75$, and they save 23.4% of the time steps to achieve the optimal performance compared with uniform sampling. These simulation results show that selecting a proper priority factor can speed up the training process, which is attributed to the priority sampling technique more effectively sampling the poorly trained data from the experience pool. Note that the optimal priority factor when $N_c = 60$ is larger than when $N_c = 40$, which shows that more prioritization is needed in a larger number of channels, because it produces more experience. Moreover, the training process is unstable when the priority factor is large, especially when $\alpha = 0.75$ and $\alpha = 1$, because the priority factor changes the state distribution of the visited experience compared with the original state distribution, which introduces estimation bias, and the solution to which the estimates will converge is changed. The larger the priority factor, the more the state distribution changes.

Table 7 shows the robustness of the DDPG scheme with different priority factors. The statistical results show that the DDPG scheme with an appropriate priority factor can remain robust as the DDPG scheme with uniform sampling, while a priority factor that is too large may cause poor performance due to the change in experience distribution.

These simulations demonstrate that the proposed prioritized sampling technique can significantly improve convergence performance. Selecting a proper priority factor makes the training process more effective.

V. CONCLUSION

In this paper, we provided a reinforcement learning-based DDPG scheme for the power control problem and proposed a prioritized sampling technique for the DDPG scheme. We considered a wireless communication scenario with one transmitter, one receiver, and a few interferers, and the total interference was modeled as MSI. The transmitter only senses the summation of interference and noise power on each channel and specifies the transmit power vector to maximize the reward performance by using a DDPG scheme. The DDPG scheme approximates a power selection policy to a CNN and approximates an evaluation of the policy to a fully connected network. We have shown that by applying a DDPG-based power control scheme, a communication user can achieve optimal reward performance without full channel CSI. Moreover, we have proposed a prioritized sampling technique which samples poor-trained experiences with a higher probability to further accelerate the learning of the DDPG scheme. The simulation results reveal that our proposed DDPG scheme for power control has a higher probability of achieving near optimal reward performance than the DQN scheme, while the training is slightly slower. This means that compared with the DQN scheme, the DDPG scheme significantly improves the robustness of learning at the expense of a small amount of training speed. At the same time, with the prioritized sampling technique, the training process of the DDPG scheme becomes more effective. In particular, by selecting a proper priority factor, the training process is accelerated by up to 36.4% compared with uniform sampling. Note that selecting a priority factor that is too large may cause agent convergence to a poor policy.

REFERENCES

[1] T. Erpek, Y. E. Sagduyu, and Y. Shi, "Deep learning for launching and mitigating wireless jamming attacks," *IEEE Trans. Cognit. Commun. Netw.*, vol. 5, no. 1, pp. 2–14, Mar. 2019.

[2] E. Ghadimi, F. D. Calabrese, G. Peters, and P. Soldati, "A reinforcement learning approach to power control and rate adaptation in cellular networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, France, May 2017, pp. 1–7.

[3] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, "A survey on jamming attacks and countermeasures in WSNs," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 4, pp. 42–56, 4th Quart., 2009.

[4] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4331–4340, Sep. 2011.

[5] K. Shen and W. Yu, "Fractional programming for communication systems—Part I: Power control and beamforming," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2616–2630, May 2018.

[6] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, no. 1, pp. 237–285, Jan. 1996.

[7] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.

[8] M. Chiang, C. Wei Tan, D. P. Palomar, D. O'Neill, and D. Julian, "Power control by geometric programming," *IEEE Trans. Wireless Commun.*, vol. 6, no. 7, pp. 2640–2651, Jul. 2007.

[9] S. G. Glisic, A. Mammela, V.-P. Kaasila, and M. D. Pajkovic, "Rejection of frequency sweeping signal in DS spread spectrum systems using complex adaptive filters," *IEEE Trans. Commun.*, vol. 43, no. 1, pp. 136–145, Jan. 1995.

[10] V. P. Mhatre, K. Papagiannaki, and F. Baccelli, "Interference mitigation through power control in high density 802.11 WLANs," in *Proc. IEEE INFOCOM-26th IEEE Int. Conf. Comput. Commun.*, Barcelona, Spain, May 2007, pp. 535–543.

[11] L. P. Qian, Y. J. Zhang, and J. Huang, "MAPEL: Achieving global optimality for a non-convex wireless power control problem," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1553–1563, Mar. 2009.

[12] C. S. Chen, K. W. Shum, and C. W. Sung, "Round-robin power control for the weighted sum rate maximisation of wireless networks over multiple interfering links," *Eur. Trans. Telecommun.*, vol. 22, no. 8, pp. 458–470, Dec. 2011.

[13] A. Gjendemsjo, D. Gesbert, G. E. Oien, and S. G. Kiani, "Binary power control for sum rate maximization over multiple interfering links," *IEEE Trans. Wireless Commun.*, vol. 7, no. 8, pp. 3164–3173, Aug. 2008.

[14] C. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992.

[15] L. Xiao, Y. Li, J. Liu, and Y. Zhao, "Power control with reinforcement learning in cooperative cognitive radio networks against jamming," *J. Supercomput.*, vol. 71, no. 9, pp. 3237–3257, Sep. 2015.

[16] S. Dzulkifly, L. Giupponi, F. Said, and M. Dohler, "Decentralized Q-learning for uplink power control," in *Proc. IEEE 20th Int. Workshop Comput. Aided Modeling Design Commun. Links Netw. (CAMAD)*, Guildford, U.K., Sep. 2015, pp. 54–58.

[17] V. Mnih, K. Kavokcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with deep reinforcement learning," in *Proc. Neural Inf. Process. Syst. (NIPS)*, Lake Tahoe, NV, USA, 2013, p. 1–9.

[18] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[19] G. Han, L. Xiao, and H. V. Poor, "Two-dimensional anti-jamming communication based on deep reinforcement learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, New Orleans, LA, USA, Mar. 2017, pp. 2087–2091.

[20] Y. Chen, Y. Li, D. Xu, and L. Xiao, "DQN-based power control for IoT transmission against jamming," in *Proc. IEEE 87th Veh. Technol. Conf. (VTC Spring)*, Porto, Portugal, Jun. 2018, pp. 1–5.

[21] L. Xiao, D. Jiang, D. Xu, H. Zhu, Y. Zhang, and H. V. Poor, "Two-dimensional antijamming mobile communication based on reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 9499–9512, Oct. 2018.

[22] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2239–2250, Oct. 2019.

[23] T. Zhang and S. Mao, "Joint power and channel resource optimization in soft multi-view video delivery," *IEEE Access*, vol. 7, pp. 148084–148097, Oct. 2019.

[24] R. Sutton, D. McAllester, and S. Singh, "Policy Gradient Methods for Reinforcement Learning with Function Approximation," in *Proc. Neural Inf. Process. Syst. (NIPS)*, Denver, CO, USA, 1999, pp. 1057–1063.

[25] N. Heess, D. Silver, and Y. W. The, "Actor-critic reinforcement learning with energy-based policies," in *Proc. Eur. Workshop Reinforcement Learn. (EWRL)*, Edinburgh, Scotland, 2012, pp. 1–11.

[26] 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," 2015, *arXiv:1509.02971*. [Online]. Available: http://arxiv.org/abs/1509.02971

[27] W. Li, J. Wang, L. Li, G. Zhang, Z. Dang, and S. Li, "Intelligent anti-jamming communication with continuous action decision for ultra-dense network," in *Proc. ICC-IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, May 2019, pp. 1–7.

[28] Y. Zhang, X. Wang, and Y. Xu, "Energy-efficient resource allocation in uplink NOMA systems with deep reinforcement learning," in *Proc. 11th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Xi'an, China, Oct. 2019, pp. 1–6.

[29] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for interference management," *IEEE Trans. Signal Process.*, vol. 66, no. 20, pp. 5438–5453, Oct. 2018.

[30] F. D. Calabrese, L. Wang, E. Ghadimi, G. Peters, L. Hanzo, and P. Soldati, "Learning radio resource management in RANs: Framework, opportunities, and challenges," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 138–145, Sep. 2018.

[31] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," 2015, *arXiv:1511.05952*. [Online]. Available: http://arxiv.org/abs/1511.05952

[32] Y. Hou, L. Liu, Q. Wei, X. Xu, and C. Chen, "A novel DDPG method with prioritized experience replay," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Banff, AB, Canada, Oct. 2017, pp. 316–321.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

SHIYANG ZHOU received the B.E. and M.E. degrees from the University of Electronic Science and Technology of China, in 2014 and 2017, respectively, where he is currently pursuing the Ph.D. degree in wireless communication engineering with the National Key Laboratory of Science and Technology on Communications. His current research interests include artificial intelligence, cognitive radio, and machine learning on communication.

YUFAN CHENG (Member, IEEE) received the B.Eng. and M.Eng. degrees in electrical engineering from the University of Electronic Science and Technology of China (UESTC), in 1994 and 1997, respectively. She is currently a Boffin with the National Key Laboratory of Science and Technology on Communications, UESTC. Her current research interests include cognitive radio and signal processing in wireless communication.

XIA LEI (Member, IEEE) received the Ph.D. degree in communication and information systems from the University of Electronic Science and Technology of China, Chengdu, China, in 2005. She is currently a Professor and a Ph.D. Supervisor with the University of Electronic Science and Technology of China. She has been involved in several projects. She has published over 40 international journals. Her research interest includes wireless broadband communication systems.

HUANHUAN DUAN received the B.S. degree from the University of Electronic Science and Technology of China, in 2018, where she is currently pursuing the master's degree. Her current research interest includes signal processing in wireless communication.