Deep Deterministic Policy Gradient (DDPG)-Based Energy Harvesting Wireless Communications

Chengrun Qiu, Yang Hu, Yan Chen, *Senior Member, IEEE*, and Bing Zeng, *Fellow, IEEE*

***Abstract*—To overcome the difficulties of charging the wireless sensors in the wild with conventional energy supply, more and more researchers have focused on the sensor networks with renewable generations. Considering the uncertainty of the renewable generations, an effective energy management strategy is necessary for the sensors. In this paper, we propose a novel energy management algorithm based on the reinforcement learning. By utilizing deep deterministic policy gradient (DDPG), the proposed algorithm is applicable for the continuous states and realizes the continuous energy management. We also propose a state normalization algorithm to help the neural network initialize and learn. With only one day's real solar data and the simulative channel data for training, the proposed algorithm shows excellent performance in the validation with about 800 days length of real solar data. Compared with the state-of-the-art algorithms, the proposed algorithm achieves better performance in terms of long-term average net bit rate.**

***Index Terms*—Deep deterministic policy gradient (DDPG), energy harvesting, Lyapunov optimization, Markov decision process (MDP), reinforcement learning, sensor networks.**

I. Introduction

Nowadays, wireless sensor networks have been widely deployed in many areas, such as environmental monitoring, traffic monitoring, and smart cities [1]. However, with the conventional energy supply, the frequency of replacing the battery will be high, which limits the deployment of wireless sensor networks, especially when the networks are located in the wild and the sensors are too small to be found. A promising solution to reduce the battery replacement and maintain the network lifetime is to apply the energy harvesting technique on the rechargeable battery of the sensors [2], [3]. Although energy harvesting wireless communications have the attractive advantages, the energy management is challenging due to the stochastic nature of the renewable generations and the wireless fading channel. Therefore, an effective energy management algorithm in energy harvesting wireless communications is necessary to improve the communication quality.

In the literature, different energy management strategies have been proposed for the energy harvesting wireless communications, and the most typical and effective ones are the water-filling algorithm, Markov decision process (MDP) and Lyapunov algorithm. Water-filling algorithm is based on the convex optimization with the Karush–Kuhn–Tucher (KKT) condition, and is only capable of optimizing the convex objective within a finite horizon. Such an algorithm is offline, which has to know all of the precise value of all states. Thus, it is not practical and is generally used to give some guidance for the design of online algorithms. Ulukus *et al.* [4] reviewed the water-filling algorithms for different reward optimizations, models, and constraints. Ozel *et al.* [5] optimized the average throughput with finite epochs of an energy harvesting communication system with the KKT condition. In detail, an offline water-filling algorithm was proposed, and based on which an online water-filling algorithm was also designed. Other water-filling algorithms such as scheduling the energy for fading multiple access channels were conducted in [6]. Zhang *et al.* [7] proposed an energy-efficient sleep scheduling in wireless body area networks.

In order to derive effective online energy management algorithms, MDP has been widely utilized in energy harvesting communications [8]–[12]. Ku *et al.* [8] optimized the long-term average net bit rate in energy harvesting communications with an energy-modulation management algorithm using MDP. In energy harvesting cooperative communications, Ku *et al.* [9] applied MDP to minimize the expected symbol error rate in one-way relay energy harvesting network with the decode-and-forward (DF) protocol, while Li *et al.* [10] used MDP to minimize the long-term average outage probability in two-way relay energy harvesting networks with both DF and amplify-and-forward (AF) protocols. Yadav *et al.* [13] presented a partially observable MDP to manage the energy in energy harvesting sensors, which shows better performance in minimizing the pairwise error. Gong *et al.* [14] proposed a framework for analyzing the energy harvesting wireless transmissions with stochastic energy harvesting rate. Ni and Motani [15] performed MDP-based online optimization to maximize the number of bits decoded for an energy harvesting receiver with a time-switching architecture. By restricting the state space, they presented a computationally efficient policy with good

Manuscript received April 27, 2019; revised May 18, 2019; accepted June 3, 2019. Date of publication June 5, 2019; date of current version October 8, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61672137 and Grant 61602090, and in part by the 111 Project under Grant B17008. The work of Y. Chen was supported by the Thousand Youth Talents Program of China. *(Corresponding author: Yan Chen.)*

C. Qiu and Y. Chen are with the Center for Intelligent Networking and Communications, University of Electronic Science and Technology of China, Chengdu 611731, China, and also with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: cr_qiu@std.uestc.edu.cn; eecyan@uestc.edu.cn).

Y. Hu and B. Zeng are with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: yanghu@uestc.edu.cn; eezeng@uestc.edu.cn).

Digital Object Identifier 10.1109/JIOT.2019.2921159

performance and low complexity. While MDP is an effective tool for designing an effective online energy management algorithm, it faces the curse of dimensionality when the number of states is large.

To overcome the shortcomings of MDP, many researchers tried to use Lyapunov optimization to optimize the long-term objectives. Unlike the MDP method, Lyapunov optimization works with the continuous state and action, i.e., discretization is not necessary. Qiu *et al.* [16]–[18] applied Lyapunov optimization theory to optimize different energy harvesting wireless networks. The results showed that Lyapunov optimization can achieve better performance compared with MDP [8], especially at the high signal-to-noise ratio (SNR) regimes. Amirnavaei and Dong [19] maximized the long-term average throughput in energy harvesting wireless networks with Lyapunov optimization. Their algorithm showed better performance than the traditional water-filling algorithm in [5]. Cui *et al.* [20] utilized Lyapunov optimization to study the delay-aware resource control problem, where the system throughput, the sum delay, and the power consumption were jointly optimized.

With the development of the reinforcement learning, more and more works have focused on solving the energy harvesting problems with the aid of reinforcement learning. Cao *et al.* [21] reviewed the state-of-the-art machine learning-based approaches in multiaccess edge computing intelligent offloading. By utilizing the Q-learning, Blasco *et al.* [22] proposed algorithms to maximize long-term expected throughput of the point-to-point energy harvesting wireless communications. Atallah *et al.* [23] used the Q-learning method to optimize the downlink traffic scheduling in vehicular networks during a discharge period. Based on SARSA, Ortiz *et al.* [24] proposed a power allocation policy to maximize the throughput at the receiver of a two-hop energy harvesting communications. However, the existing works that applied the model-free reinforcement learning framework to solve the stochastic optimization problems all have to discretize the continuous variables of the state and action scenario into a finite set of discrete values (levels). For example, the state and action have to be discretized in Q-learning and SARSA while the action interval discretization is necessary in deep Q-learning (DQN). Since there are some shortcomings similar to that of the MDP, researchers began to use the actor–critic algorithm in reinforcement learning. Aoudia *et al.* [25] used actor–critic algorithm to manage the energy to maximize the quality of service while avoiding power failures for energy harvesting wireless sensor networks. Wei *et al.* [26] investigated the optimal policy for user scheduling and resource allocation in HetNets powered by hybrid energy with the purpose of maximizing energy efficiency of the overall network.

In this paper, we investigate continuous energy management to maximize the net bit rate in energy harvesting wireless communications using DDPG [27]. Instead of optimizing the long-term throughput based on Shannon capacity, we try to deploy DDPG for maximizing the average net bit rate which is a nonconvex problem. Due to the nonconvex characteristic, the existing methods formulate the problem as an MDP. However, considering the curse of dimensionality in MDP, we propose to solve this nonconvex problem with the aid of reinforcement learning.

DDPG is an improved version of the actor–critic algorithm, which uses the deep neural networks to estimate the optimal policy function instead of choosing the action based on a specific distribution. On the one hand, DDPG can avoid the curse of dimensionality compared with MDP and Q-learning which require the discretization of the state. On the other hand, DDPG can solve the nonconvex objective function in a long-term average form. Therefore, it is attractive to study whether DDPG can resolve the continuous energy management problem in energy harvesting wireless communications. The major contributions of this paper are shown as follows.

1) We formulate the energy management problem in the point-to-point wireless communications and the one-way relay cooperative communications as a reinforcement learning problem to maximize the long-term average net bit rate. To resolve the optimization problem with DDPG, we propose to rewrite the problem into a state, action, and reward form.
2) DDPG based on the actor–critic algorithm is used to generate an optimal policy with continuous states and actions. Specifically, the actor net in DDPG is responsible for learning the optimal policy for continuous actions, while the critic net maintains on fitting the real Q-table for preventing the curse of dimensionality. We use some activation functions in deep neural networks to help generate nonlinear characteristics of the net bit rate. We discuss in detail the training and validation processes of the proposed algorithm. Moreover, in order to achieve better results, we propose a state normalization to preprocess the input. We also theoretically derive the time and space complexity of the training and the validation processes.
3) Simulations are conducted to evaluate the performance of the proposed policy. To train the proposed policy, we use one day's data for training. The data includes the solar irradiance measured in real world and the Rayleigh fading channel generated by Jakes model. Then we repeat the three months' data for several times to validate our policy's performance. The results show that, compared with the state-of-the-art algorithms, our trained policy has better performance especially at the low SNR region.

Notice that besides formulating the continuous energy management problem in energy harvesting wireless communication as a reinforcement learning problem and applying DDPG to find the solution, in this paper, we also introduce a compact neural network structure for DDPG and utilize the state normalization to help the neural network initialize and learn. These procedures may be helpful in solving other wireless communication problems with reinforcement learning.

The rest of this paper is organized as follows. In Section II, we briefly review DDPG. Then, the system models of energy harvesting wireless networks are introduced in Section III. The energy management policy using DDPG is proposed in detail in Section IV. Section V shows the simulation results and Section VI concludes this paper.

II. Brief Review of DDPG

DDPG is a reinforcement learning framework that can handle the continuous action spaces based on the actor–critic algorithm. Based on policy gradient and DQN, DDPG is capable of solving many problems with continuous action spaces, such as the tasks in gym[1] and TORCS. DDPG can directly use the raw observations to learn, and used fewer steps of experience of learning than DQN in the Atari domain [27].

DDPG has two main networks: 1) the critic net and 2) the actor net. Both the critic net and the actor net contains two subnets: 1) the online net and 2) the target net, whose architectures are the same. These four neural networks are composed of various layers, and all layers contain their corresponding parameters. All parameters in a specific network are denoted as θ. The critic net is trained to simulate the real Q-table using neural networks without the curse of dimensionality. The actor net is trained for generating a deterministic policy instead of the policy gradient which chooses a random action from a determined distribution.

In policy gradient, the agent's behavior a is determined by π, which maps states to a probability distribution over the actions. Given the instantaneous state s_t and the action a_t, if the action's policy is deterministic, denoted as μ, we can avoid the inner expectation and write the Q-table as

$$Q^{\mu}(s_t, a_t) = \mathbb{E}_{r_t, s_{t+1} \sim \Psi}\Big[r(s_t, a_t) + \gamma\big[Q^{\mu}(s_{t+1}, \mu(s_{t+1}))\big]\Big] \quad (1)$$

where $r(s_t, a_t)$ represents the reward of the state s_t and the action a_t, γ stands for the discount factor in Bellman equation, and Ψ is the corresponding expectation distribution for s_{t+1} and r_t.

When the deterministic policy μ is generated from a randomly initialized stochastic policy ψ, with the approximative Q-table parameterized by θ^Q, the loss of the critic net is defined to measure the distance between the two side of the Bellman equation, which can be expressed as

$$L(\theta^Q) = \mathbb{E}_{s_t \sim \rho^{\psi}, a_t \sim \psi, r_t \sim \Psi}\Big[\Big(Q(s_t, a_t|\theta^Q) - y_t\Big)^2\Big] \quad (2)$$

where ρ^{ψ} represents the distribution of the state s_t under the current deterministic policy ψ, and θ^Q can be considered as the variables in deep Q-network and y_t is defined as follows:

$$y_t = r(s_t, a_t) + \gamma Q\Big(s_{t+1}, \mu(s_{t+1})|\theta^Q\Big). \quad (3)$$

The actor net updates the policy with the aid of the critic net, where the policy's updating gradient can be written as follows:

$$\nabla_{\theta^{\mu}} J \approx \mathbb{E}_{s_t \sim \rho^{\psi}}\Big[\nabla_a Q\Big(s, a|\theta^Q\Big)|_{s=s_t, a=\mu(s_t)} \nabla_{\theta^{\mu}} \mu(s|\theta^{\mu})|_{s=s_t}\Big] \quad (4)$$

where θ^{μ} can be considered as the variables of the online actor net.

The procedure of an entire training process can be described as follows. First, with the action $\mu(s_t)$ given by the actor net after the previous training, DDPG adds some noise n_t and generates the action $a_t = \mu(s_t) + n_t$. Then with the action a_t working on the environment, DDPG can get a reward r_t and a next state s_{t+1}. DDPG will store the set (s_t, a_t, r_t, s_{t+1}) in the experience replay buffer. After that, DDPG randomly chooses N sets in the buffer to make up a mini-batch and inputs it to both the actor net and the critic net. With the mini-batch, the target net of the actor net outputs the action $\mu'(s_{i+1})$ with regard to $\theta^{\mu'}$ to the critic net. With the mini-batch and $\mu'(s_{i+1})$, the target net of the critic net can calculate y_i based on (3) and input it to the online net.

With a given optimizer, e.g., Adamoptimizer, the critic net will update its own online net. Afterwards, the actor online net gives the mini-batch action $a = \mu(s_i)$ to the critic online net to achieve the action a's gradient $\nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)}$. With its own optimizer, the parameter θ^{μ}'s gradient $\nabla_{\theta^{\mu}} \mu(s|\theta^{\mu})|_{s=s_i}$ can be derived. With the above two gradients, the actor net can update the actor online net with the following approximation:

$$\nabla_{\theta^{\mu}} J \approx \frac{1}{N}\sum_i \Big[\nabla_a Q\Big(s, a|\theta^Q\Big)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta_{\mu}} \mu(s|\theta^{\mu})|_{s=s_i}\Big]. \quad (5)$$

Finally, DDPG softly updates the target nets in both critic net and actor net with a small constant τ, i.e.,

$$\begin{aligned} \theta^{Q'} &\leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'} \\ \theta^{\mu'} &\leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'}. \end{aligned} \quad (6)$$

III. System Models of Energy Harvesting Wireless Networks

As shown in Fig. 1, in this paper, we consider the energy management for two energy harvesting networks: the point-to-point network [8] and one-way relay network [9]. In the point-to-point network, the transmitter with energy harvesting capability sends the packets to the destination with the energy in the battery while the energy harvester keeps collecting the renewable generations and stores them in the battery. In particular, the energy harvested at present is only available in the subsequent periods, i.e., the harvest-store-use (HSU) model is used in this paper [3]. Thus, the energy in the battery can be written as follows:

$$b_{t+1} = \min\{b_t - \omega_t + E_{H,t}, b_{\max}\} \quad (7)$$

where t is the time index, $E_{H,t}$ is the collected energy, $\omega_t \in [0, b_t]$ means the consumed energy, and $b_{\max}$ represents the battery capacity.

In the one-way relay network, there exist two phases in an entire transmission period. In the first phase, the source broadcasts the packets to the relay and the destination, and the relay decodes the packets. In the second phase, if the decoding in the first phase succeeds, the relay transmits the re-encoded packets to the destination. Different from the point-to-point network, the energy harvester collects the energy in both two phases but only consumes the energy in the second phase. In

[1] Gym is a toolkit for developing and comparing reinforcement learning algorithms. It supports teaching agents everything from walking to playing games like Pong or Pinball.

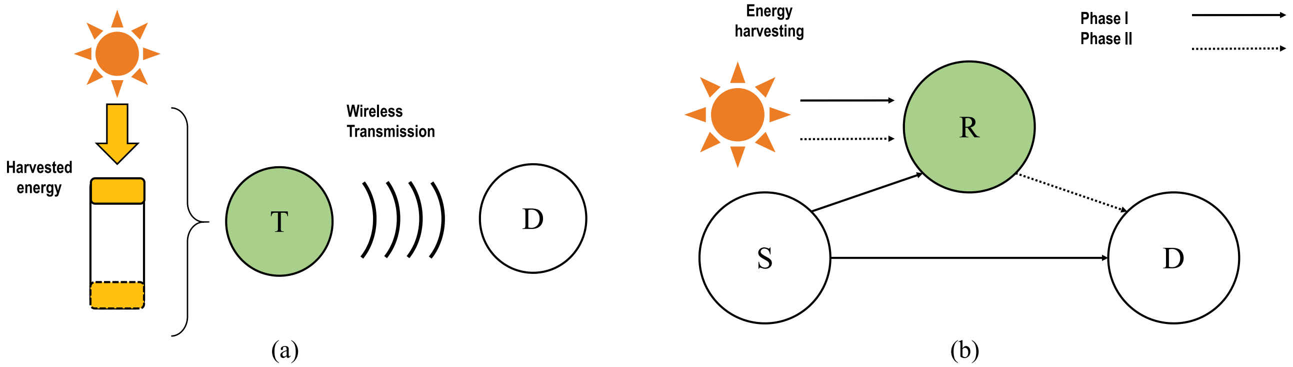


Fig. 1. Energy harvesting wireless networks. (a) Point-to-point network. (b) One-way relay network.

both networks, the channels are assumed to be independent identically distributed with Rayleigh fading.

In this paper, we use the net bit rate, which represents the expected good bits per packet transmission [8], as the system objective. The net bit rate can be influenced by many factors, including the number of bits per packet, the packet error rate, and the rate of sending packages. According to [8], the instantaneous net bit rate can be written as follows:

$$R(\zeta_t, \omega_t, b_t) = \begin{cases} \frac{\chi_m L_S}{T_p}(1-P_e)^{\chi_m L_S}, & \omega \neq 0 \\ 0, & \omega = 0 \end{cases}$$
$$\text{s.t.} \quad 0 \le \omega_t \le b_t \tag{8}$$

where χ_m, L_S, T_p, P_e, and ζ_t represent the bit number per symbol, the symbol number per packet, the packet transmission duration, the bit error rate, and the instantaneous channel power, respectively.

Equation (8) means the number of correct bits received per unit time. It is mainly determined by the bit error rate P_e and the number of bits transmitted per unit time, i.e., $(\chi_m L_S/T_p)$. With the bit error rate, the probability of one packet transmission without any bit error is $(1-P_e)^{\chi_m L_S}$. Then, we can calculate the net bit rate by multiplying the decoding success probability with the total number of bits per unit time. The condition with $\omega = 0$ is written separately because when a transmitter turned off, the transmission will stop.

The bit error rate P_e is determined by the SNR, which is different for different networks. In the point-to-point network, we choose an approximation of the bit error rate as in [8], which can be expressed as

$$P_e(\zeta_t, \omega_t, b_t) = \sum_r \alpha(m, r) \cdot \frac{1}{2}\text{erfc}\left(\sqrt{\frac{\beta(m, r)\omega_t\zeta_t}{2N_0 T_L}}\right)$$
$$\text{s.t.} \quad 0 \le \omega_t \le b_t \tag{9}$$

where erfc(·) represents complementary error function, N_0 is the noise power, (ζ_t/N_0) stands for the instantaneous channel-to-noise ratio, $\alpha(m, r)$ and $\beta(m, r)$ are two parameters related to the modulation m, which are shown in Table I [28], and r represents the specific constants determined by modulation m. The instantaneous SNR of the point-to-point network is written as $(\omega_t\zeta_t/N_0 T_L)$ in (9), which significantly affects the bit error rate P_e.

TABLE I
MODULATION RELATED PARAMETERS

Modulation schemes	Parameters$(\alpha(m,r),\beta(m,r))$
QPSK	$(\alpha(m,0),\beta(m,0)) = (1,1)$
8PSK	$(\alpha(m,0),\beta(m,0)) = (\frac{2}{3}, 2sin^2(\frac{\pi}{8}))$ $(\alpha(m,1),\beta(m,1)) = (\frac{2}{3}, 2sin^2(\frac{3\pi}{8}))$
16QAM	$(\alpha(m,0),\beta(m,0)) = (\frac{3}{4}, \frac{1}{5})$ $(\alpha(m,1),\beta(m,1)) = (\frac{1}{2}, \frac{9}{5})$

Similarly, based on the SNR in the one-way relay network with the DF protocol, the corresponding bit error rate can be written as

$$P_e(\zeta_t, \omega_t, b_t) = \begin{cases} \sum_r \alpha(m, r)\frac{1}{2}\text{erfc}\left(\sqrt{\frac{\beta(m,r)[(\omega_t/T_L)\zeta_{rd,t}+\Psi_s\zeta_{sd,t}]}{2N_0}}\right) \\ \qquad d = 1 \\ \sum_r \alpha(m, r)\frac{1}{2}\text{erfc}\left(\sqrt{\frac{\beta(m,r)\Psi_s\zeta_{sd,t}}{2N_0}}\right) \\ \qquad d = 0 \end{cases}$$
$$\text{s.t.} \quad 0 \le \omega_t \le b_t \tag{10}$$

where $d = 1$ corresponds to the case where the decoding in the relay is successful, Ψ_s represents the transmission power of the source node, and ζ_{sd} and ζ_{rd} are the channel power of the source-to-destination (SD) link and the relay-to-destination (RD) link, respectively. The decoding condition d is determined by the SNR of the source-to-relay link, and d is equal to 1 only if $(\Psi\zeta_{sr}/N_0) \ge \mathfrak{T}$, where $\mathfrak{T}$ is a constant standing for the decoding capacity threshold. The SNR of the one-way relay network is determined by both of the relay route and the source route, which is written as $([(\omega/T_L)\zeta_{rd} + \Psi_s\zeta_{sd}]/N_0)$ in (10).

IV. ENERGY MANAGEMENT WITH DDPG

In this section, we discuss how to utilize DDPG to train a stochastic energy management policy. In detail, we introduce the problem formulation, illustrate the procedures of training and validating, discuss the state normalization as data preprocessing, and analyze the complexity of the training and validation.

A. Problem Formulation

The system objective in this paper is to maximize the long-term average net bit rate in the point-to-point network and the one-way relay network, i.e.,

$$\max_{\omega_t} \lim_{T\to+\infty} \frac{1}{T}\sum_{t=0}^{T-1} R(\zeta_t, \omega_t, b_t)$$
$$\text{s.t. } 0 \le \omega_t \le b_t \tag{11}$$

where the bit error rate in the long-term objective function is defined in (9) and (10) for the point-to-point network and the one-way relay network, respectively.

In DDPG, the neural networks are trained from the set (s_t, a_t, r_t, s_{t+1}). Therefore, we have to define the state and action sets of our problem for DDPG. Moreover, with the constraint of the energy for transmission, we have to define a different action for DDPG instead of directly deploying the ω_t. In the following, we describe in detail the state set, the action set, and the reward.

1) Set of States s_t: In the point-to-point communications, the states needed to be considered in a management period include b_t, ζ_t, and $E_{H,t}$, where b_t determines the maximum energy the transmitter node can consume, ζ_t influences the bit error rate directly, and $E_{H,t}$ affects b_{t+1}. Therefore, for the point-to-point network, the sets are $s_t = (b_t, \zeta_t, E_{H,t})$.

On the other hand, there exist three independent fading channels in the one-way relay network. Nevertheless, the states contain only the channel power of the SD link $\zeta_{sd,t}$ and the channel power of the RD link $\zeta_{rd,t}$, i.e., $s_t = (b_t, \zeta_{sd,t}, \zeta_{rd,t}, E_{H,t})$. The reason of ignoring the channel power of the SR link $\zeta_{sr,t}$ is that $\zeta_{sr,t}$ only influences the decoding condition. In the training process, if the decoding fails in the first phase, the training will be directly skipped. In the validation process, if the decoding fails in the first phase, the relay will not manage the energy for transmission. The SR link only influences whether the relay is on or off, but has no impact on the energy management.

2) Set of Actions a_t: In existing energy management methods, the action is determined by the remaining energy in the battery, i.e.,

$$\forall t > 0, a_t \in [0, b_t] \tag{12}$$

or

$$\forall t > 0, a_t \in \{0, P_U, 2P_U, \ldots, nP_U\} \tag{13}$$

with $n = \lfloor b_t/P_U \rfloor$, and P_U stands for the predefined basic energy quantum for transmission.

In this paper, we set the continuous possible action $a_t \in [0, 1]$ since the action in DDPG has to be bounded by a constant. With such an action set, the final energy consumed for transmission is $a_t \times b_t$, which can guarantee that the consumed energy will not exceed the remaining energy in the battery. Thus, the action in this paper can be written as

$$a_t = \mu((b_t, \zeta_t, E_{H,t})|\theta^{\mu}) \tag{14}$$

or

$$a_t = \mu((b_t, \zeta_{sd,t}, \zeta_{rd,t}, E_{H,t})|\theta^{\mu}) \tag{15}$$

with the range of [0, 1].

Algorithm 1 State Normalization

```
Input:
    All instantaneous variables needed to be normalized: $b_t$ ,$\zeta_t$ ($\zeta_{sd,t}$, $\zeta_{rd,t}$) and $E_{H,t}$;
    Scale factors: $\lambda_1$, $\lambda_2$;
    Means and standard deviations of the variables: $\eta_{E_H}$, $\eta_\zeta$, $\sigma_{E_H}$ and $\sigma_\zeta$;
Output:
    Normalized variables $\hat{b}_t$, $\hat{\zeta}_t$ ($\hat{\zeta}_{sd,t}$, $\hat{\zeta}_{rd,t}$) and $\hat{E}_{H,t}$
$\hat{b}_t = \frac{b_t}{\lambda_1}$
    $\hat{\zeta}_t = \frac{\zeta_t - \eta_\zeta}{\sigma_\zeta}$
    $\hat{E}_{H,t} = \frac{E_{H,t} - \eta_{E_H}}{\lambda_2 \sigma_{E_H}}$
return $\hat{b}_t$, $\hat{\zeta}_t$ ($\hat{\zeta}_{sd,t}$, $\hat{\zeta}_{rd,t}$) and $\hat{E}_{H,t}$
```

3) Reward: With the Q-table in Q-learning, the long-term average net bit rate can be written using the Bellman equation as follows:

$$Q^{\mu}(s_t, a_t) = \mathbb{E}_{r_t, s_{t+1}\sim E}\Big[R(s_t, a_t) + \gamma\big[Q^{\mu}(s_{t+1}, \mu(s_{t+1}))\big]\Big] \tag{16}$$

where the reward $R(s, a)$ is chosen as the corresponding net bit rate in (8). Now the problem is to find an action to maximize the Q-value in (16) with DDPG.

B. State Normalization

In deep learning, the distribution of each layer's inputs keeps changing, which slows down the training by requiring lower learning rates and careful parameter initialization. Ioffe and Szegedy proposed the batch normalization in [29] to allow the training to use much higher learning rates and relaxed initialization. Similar to the batch normalization, we propose a state normalization to preprocess the training sample states for a much easier and faster training.

The three variables b_t, ζ_t, and $E_{H,t}$ in the state set may lie in different ranges, which may cause problem in the training process. To prevent such a problem, we normalize the variables b_t, ζ_t, and $E_{H,t}$ separately. The state normalization is shown in Algorithm 1, where we use two extra scale factors in the normalization. The reason can be explained as follows. According to (7), the energy in the battery is in the form of a queue and it will be influenced by the action. In such a case, it is difficult to use a constant value to approximate the average energy in the battery. Thus, we scale down the remaining energy in the battery of all epoches. We also scale the value of the normalized $\hat{E}_{H,t}$ to adjust the balance between the influence of $E_{H,t}$ and b_t.

C. Training and Validating Algorithms

In the training process, the parameters of the critic net to approach the Q-table and the parameters of the actor net to train the policy are updated continuously. The training algorithm based on DDPG is shown in Algorithm 2.

With the trained $\mu(s|\theta^{\mu})$ from Algorithm 2, we can derive the corresponding algorithm for energy management in the

Algorithm 2 Energy Management Policy Training Algorithm

```
Input:
    DDPG parameters:
    Training epoch length $T$, training sample length $L$;
    critic learning rate $lr_c$, actor learning rate $lr_a$;
    discount factor $\gamma$, soft update factor $\tau$;
    replay buffer $B$, mini-batch size $N$;
    Gaussian distributed behavior noise $n$ with the average value $n_0$, behavior noise decay factor $\kappa$;
    scale factors in state normalization: $\lambda_1$, $\lambda_2$
    EH communication training samples:
    Solar energy data, fading channel power data, modulation, average channel-to-noise ratio
Initialize: Empty replay buffer, randomly initialized actor net and critic net, initialized state $s_0$
$\eta_\zeta = \frac{1}{L}\sum_{i=0}^{L}\zeta_i$, $\eta_{E_H} = \frac{1}{L}\sum_{i=0}^{L}E_{H,i}$
$\sigma_\zeta^2 = \frac{1}{L}\sum_{i=0}^{L}(\zeta_i - \eta_\zeta)^2$, $\sigma_{E_H}^2 = \frac{1}{L}\sum_{i=0}^{L}(E_{H,i} - \eta_{E_H})^2$
for each epoch t=0,1,2,...,$T$ do
    for each sample $l$=0,1,...,$L$ do
        State normalize the current state $s_l$ to $\hat{s}_l$
        Get the action with the actor net and the behavior noise:
        $a_l = \min(\max(\mu(\hat{s}_l|\theta^\mu) + n_l, 0), 1)$
        Let $\omega_l = a_l b_l$ and get $r_l$ and $b_{l+1}$ from equation (7) and (8)
        Let $s_{l+1}$ includes $b_{l+1}$, $E_{H,l+1}$ and $\zeta_{l+1}$ and state normalize it into $\hat{s}_{l+1}$.
        if the replay buffer is not full then
            Store the set $(\hat{s}_l, a_l, r_l, \hat{s}_{l+1})$ in the buffer
        else
            Randomly replace a set in the buffer with $(\hat{s}_l, a_l, r_l, \hat{s}_{l+1})$
            $n_0 = n_0 \times \kappa$
            Randomly choose N sets to form a mini-batch: $(\hat{s}_i, a_i, r_i, \hat{s}_{i+1})$, $\forall i = 1, 2, ..., N$
            $y_i = r_i + \gamma Q'(\hat{s}_{i+1}, \mu'(\hat{s}_{i+1})|\theta^{Q'})$
            Update the critic online net by minimizing the loss:
            $L(\theta^Q) = \frac{1}{N}\sum_i \left((Q(\hat{s}_i, a_i|\theta^Q) - y_i)^2\right)$
            Update the actor online net by the sampled policy gradient (5)
            Soft update the target network (6)
        end if
    end for
end for
return The online actor net $\mu(\hat{s}|\theta^\mu)$
```

testing process as shown in Algorithm 3. Note that since the network is trained with the normalized state, in the testing process, we also have to preprocess the input state.

D. Complexity Analysis

From the above discussions, we can see that the training algorithm includes the normalization, the replay buffer and four neural networks, while the validation algorithm is only made up of the normalization and the online actor net. In the following, we will derive the time complexity (computations) with regard to floating point operations per second (FLOPS) and space complexity (memory) of the training and validating algorithms, respectively.

Algorithm 3 Energy Management Policy Validating Algorithm

```
Input:
    Current states: $b$, $\zeta$ $(\zeta_{sd}, \zeta_{rd})$ and $E_H$;
    Trained online actor net $\mu(s|\theta^\mu)$
    Parameters in state normalization: $\eta_\zeta$ $(\eta_{\zeta_{sd}}, \eta_{\zeta_{rd}})$, $\eta_{E_H}$, $\sigma_\zeta$ $(\sigma_{\zeta_{sd}}, \sigma_{\zeta_{rd}})$, $\sigma_{E_H}$, $\lambda_1$ and $\lambda_2$
Output:
    Action $a$
while Running do
    $\hat{b} = \frac{b}{\lambda_1}$
    $\hat{\zeta} = \frac{\zeta - \eta_\zeta}{\sigma_\zeta}$
    $\hat{E}_H = \frac{E_H - \eta_{E_H}}{\lambda_2 \sigma_{E_H}}$
    $a = \mu(\hat{s}|\theta^\mu)$
    return $a$
end while
```

1) Training: The state normalization is conducted at every epoch of the training because without action we cannot know the value of b_t. Thus, the time complexity of state normalization is $\mathcal{N}(s)$, where $\mathcal{N}(s)$ is the number of the variables in the state set. The space complexity is related to the number of the variables in the state, i.e., $2\mathcal{N}(s)$ because the algorithm has to record the means and standard deviations to avoid repeated calculation. The experience replay buffer in DDPG occupies some space to store the state sets, hence the space complexity is N.

Since the input state of the energy harvesting communications is different from that of the image/video, there is no convolution layer in both the actor net and the critic net. For dot products of a P vector and an $P \times Q$ matrix, the FLOPS computations is $(2P-1)Q$ because for every column in matrix we need to multiply P times and add $P-1$ times.

We also have to derive the computations of activation layers. When calculating FLOPS, we usually count addition, subtraction, multiplication, division, exponentiation, square root, etc., as a single FLOP. The computations is Q with Q inputs for Relu layers, $4 \times Q$ for sigmoid layers, and $6 \times Q$ for tanh layers.

Assuming that the actor net contains J fully connected layers and the critic net contains K fully connected layers, considering the bias adding in fully connected layers the time complexity can be calculated as

$$\begin{aligned} & v_{\text{activation}} u_i + 2 \times \sum_{j=0}^{J-1} u_{\text{actor},j} u_{\text{actor},j+1} \\ & + 2 \times \sum_{k=0}^{K-1} u_{\text{critic},k} u_{\text{critic},k+1} \\ & = O\left(\sum_{j=0}^{J-1} u_{\text{actor},j} u_{\text{actor},j+1} + \sum_{k=0}^{K-1} u_{\text{critic},k} u_{\text{critic},k+1}\right) \quad (17) \end{aligned}$$

where u_i means the unit number in the ith layer, u_0 equals the input size, and $v_{\text{activation}}$ means the corresponding parameters determined by the type of the activation layer.

For a fully connected layer, there is a $P \times Q$ matrix and a Q bias vector. Hence, the memory of one fully connected layer is $(P+1)Q$. Because the activation do not need saved weights, the space complexity of the neural networks is formulated as

$$\sum_{j=0}^{J-1}(u_{\text{actor},j}+1)u_{\text{actor},j+1}+\sum_{k=0}^{K-1}(u_{\text{critic},k}+1)u_{\text{critic},k+1}$$
$$=O\left(\sum_{j=0}^{J-1}u_{\text{actor},j}u_{\text{actor},j+1}+\sum_{k=0}^{K-1}u_{\text{critic},k}u_{\text{critic},k+1}\right). \quad (18)$$

Therefore, the overall time complexity of our training algorithm is

$$2\times\sum_{j=0}^{J-1}u_{\text{actor},j}u_{\text{actor},j+1}+2\times\sum_{k=0}^{K-1}u_{\text{critic},k}u_{\text{critic},k+1}$$
$$+\ v_{\text{activation}}u_i+\mathcal{N}(s)$$
$$=O\left(\sum_{j=0}^{J-1}u_{\text{actor},j}u_{\text{actor},j+1}+\sum_{k=0}^{K-1}u_{\text{critic},k}u_{\text{critic},k+1}\right)$$
$$+\ O(\mathcal{N}(s)) \quad (19)$$

and the overall space complexity of our training algorithm is

$$\sum_{j=0}^{J-1}(u_{\text{actor},j}+1)u_{\text{actor},j+1}+\sum_{k=0}^{K-1}(u_{\text{critic},k}+1)u_{\text{critic},k+1}$$
$$+\ 2\times\mathcal{N}(s)+N$$
$$=O\left(\sum_{j=0}^{J-1}u_{\text{actor},j}u_{\text{actor},j+1}+\sum_{k=0}^{K-1}u_{\text{critic},k}u_{\text{critic},k+1}\right)$$
$$+\ O(\mathcal{N}(s))+O(N). \quad (20)$$

2) Validating: Since the critic net is generated to help the actor net have a faster and easier training, there is no critic net and replay buffer in the validation process. Only the state normalization and the online net in the actor net is needed. Therefore, the time complexity of the validation algorithm is

$$O\left(\sum_{j=0}^{J-1}u_{\text{actor},j}u_{\text{actor},j+1}\right)+O(\mathcal{N}(s)) \quad (21)$$

and the space complexity is the same

$$O\left(\sum_{j=0}^{J-1}u_{\text{actor},j}u_{\text{actor},j+1}\right)+O(\mathcal{N}(s)). \quad (22)$$

V. Numerical Simulations

In this section, we conduct multiple simulations to evaluate the proposed algorithms. We start the section by presenting the architecture of the actor and critic nets and listing all of the parameters in DDPG and energy harvesting communications. Then, we conduct the parametric analysis to show the influence of the parameters in DDPG on the training process and results. Finally, our optimal policy trained by DDPG is evaluated by comparing with the Lyapunov optimization [16], the MDP algorithm [8], the original actor–critic reinforcement learning [30], and the Greedy algorithm where all of the remaining energy in the battery is consumed in all periods.

TABLE II
DDPG Architecture (The Complexity Is Evaluated With FLOPs, i.e., the Number of Floating-Point Multiplication-Adds)

Net	Layer	Units	Activation	Computations	Memory
Actor	Fully connected	60			
	Fully connected	30			
	Fully connected	1	Sigmoid	4.02K	2.10K
Critic	Fully connected	60	RELU		
	Fully connected	60			
	Fully connected	60	Tanh		
	Fully connected	60			
	Fully connected	60	RELU		
	Fully connected	1		29.88K	15.00K

A. Energy Harvesting Communications Setup

In our simulations, we use the real solar power data collected in every 5 min from 7 A.M. to 5 P.M. in June from 2010 to 2012 [31]. The solar panel size is 4 cm^2 and the energy conversion efficiency is assumed to be 20%. The Rayleigh fading channel is generated with the aid of Jakes model, which can guarantee that the channel varies in a smooth way. We assume that each packet contains 1000 symbols ($L_S = 1000$) and the packet duration $T_P = 0.01$ s. One management period is set as 5×60 s, while for the one-way relay energy harvesting communication the length of one phase is 150 s. Additionally, for the one-way relay energy harvesting network, the extra settings are shown as follows. The source node's transmission power is set as 40 mW and the decoding capacity threshold $\mathfrak{T}$ is set as 15 dB. Finally, the SNR of the SR channel is limited to 40 dB, independent from the other two channels.

B. DDPG Setup

The architecture of the actor net and the critic net is shown in Table II. In DDPG, we use three fully connected layers to build the online subnet and the target subnet for the actor net, while double fully connected layers are used to generate the critic net. In the actor net, we only use the sigmoid activation to ensure that our final output action is bounded by 0 and 1. For the critic net, aside from more layers and units, we also add two Relu activation layers and one tanh activation layer. This is because that the net bit rate is a complex nonlinear function which includes the complementary error function (erfc). Hence, more layers and activations are helpful to approach the nonlinear Q-table. We only use three layers in the actor net mainly for reducing the complexity of the energy management policy, especially in real applications.

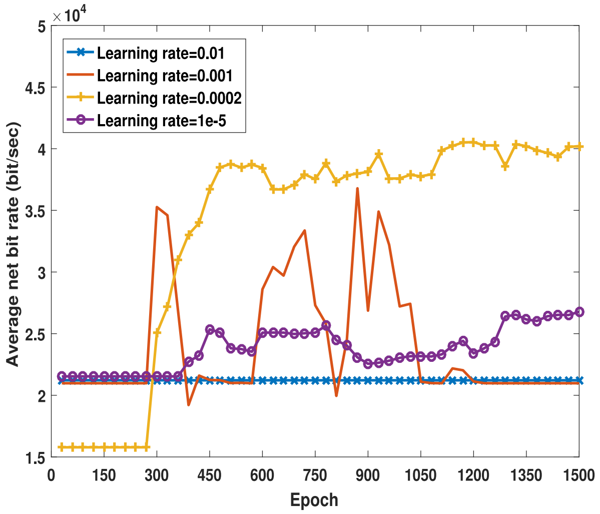


Fig. 2. Convergence of training P2P policy with different learning rate.

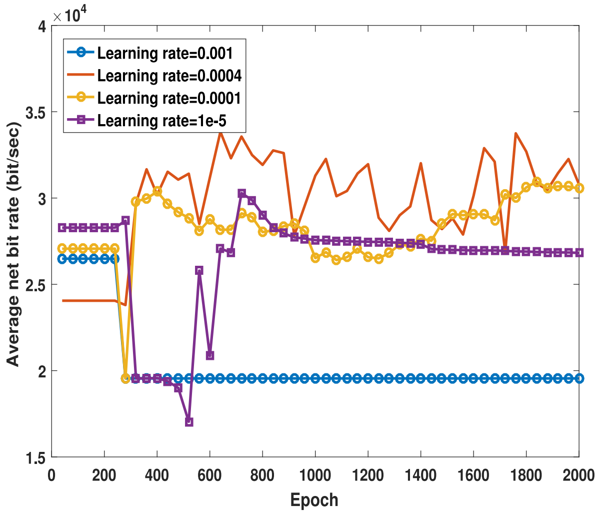


Fig. 3. Convergence of training one-way relay policy with different learning rate.

The parameter settings of DDPG are shown as below. For the point-to-point network, the training sample number is 120, where we only use the solar data of June 1st for training. The length of the training epochs is 2000, the replay buffer's capacity is 40 000 and the size of the mini-batch is 80. The learning rate of the actor net and the critic net are both set to be 2×10^{-4}. The discount factor γ of the Q-table is 0.999 and the target subnet soft update factor τ is set to be 0.01. With the behavior noise, the initialized noise's average value is set as 10 and the noise's decay factor κ is set as 0.9995. Finally, the scale factors for the state normalization λ_1 and λ_2 are set as 100 and 2, respectively.

For the one-way relay network, the length of the training sample is 240, where we apply the solar data of June 1st and June 2nd. We use more training samples since there exists a probability of the failed decoding. If the decoding fails, the policy in energy harvesting relay cannot be trained. The length of the training epochs is 1500, the learning rates of the actor and critic net are both 4×10^{-4}, and the discount factor γ in Q-table is set as 0.9. The other parameters are the same as those of DDPG in the point-to-point network.

C. Parametric Analysis

The influence of the learning rate of DDPG can be inferred from Figs. 2 and 3, where the modulations are both QPSK, the channel-to-noise ratio in the peer-to-peer network is -10 dB and the SNR in the one-way relay network is 4 dB. The learning rates of the critic net and the actor net are assumed the same.

From Fig. 2, we can see that when the learning rate is 0.01, there is no advantage in the training since the policy turns to a greedy algorithm in this case. With a learning rate of 0.001, the result fluctuates with the development of the epochs. If the learning rate goes smaller, e.g., 0.0002, the result generally maintains improving with slight fluctuations. When the learning rate becomes 10^{-5}, we can see that the result grows slowly but still with slight fluctuations.

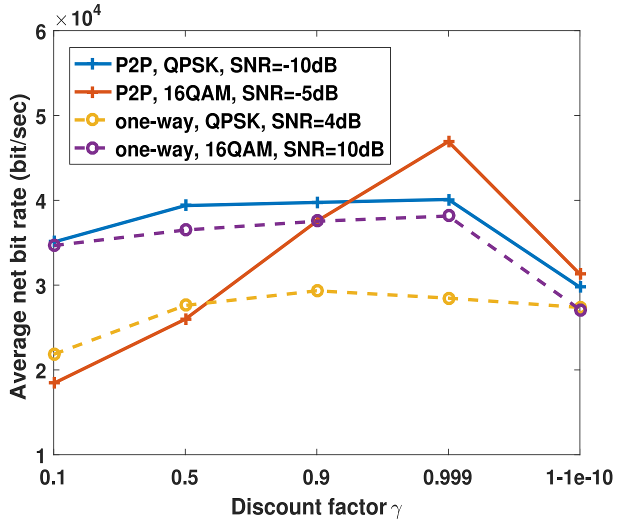


Fig. 4. Influence of the discount factor γ.

In Fig. 3, we observe similar phenomenon as that in Fig. 2. If the learning rate is too large, the average net bit rate will quickly saturate at a bad value. In a reasonable range, with a higher learning rate, the result can grow faster but with larger fluctuations. On the other hand, if the learning rate is smaller, fluctuations will be reduced at the sacrifice of the speed of the performance growth. Therefore, the learning rate should be selected properly, neither too large nor too small. Compared with Fig. 2, we find that it is harder for DDPG to converge at a good result in the one-way relay network. This is because we artificially make the relay off if the decoding d in (11) is 0. This may make the state input in DDPG not continuous in time and cause more difficulty in training a good action.

The influence of the discount factor γ is shown in Fig. 4. For both peer-to-peer network and one-way relay network, the trained energy management policy has the best performance where the discount factor is 0.9 or 0.999. If γ is too large or too small, the performance degrades. The reason can be

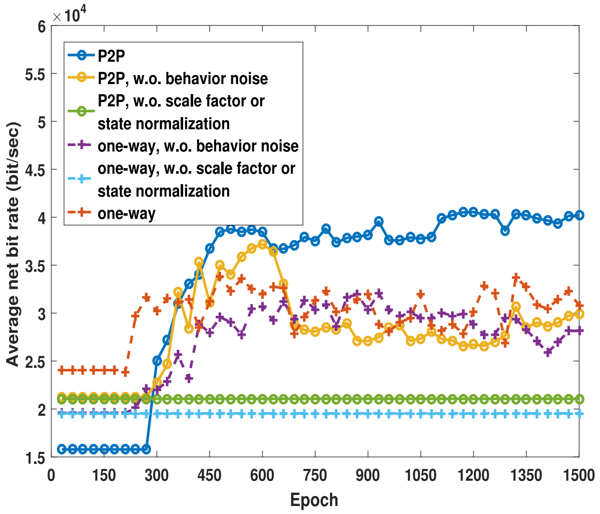


Fig. 5. Converge of training without behavior noise or state normalization.

presented as follows. In the formulation of the Bellman equation, an exact expression for the long-term average is with $\gamma = 1$. Therefore, if the gamma is too small, e.g., 0.1 or 0.5 in the figure, the trained policy will pay more attention to the instantaneous reward instead of managing the long-term energy. On the other hand, the result can still be worse if $\gamma = 1 - 1e - 10$. This is because a one-day's data cannot stand for the long-term behavior perfectly. With a γ extremely close to 1, the Q-table will consider the one-day's data as the long-term data, thus leading to a bad generalization ability of different day's solar power. Therefore, a suitable value of γ will improve the final performance of our trained policy.

Fig. 5 shows the final results of the trained policy if behavior noise or state normalization is not utilized in DDPG training. We can see from the figure that if the policy is trained without the behavior noise, in both the peer-to-peer network and the one-way relay network, the average net bit rate performance degrades. Compared with the behavior noise, from Fig. 5 we can see that the state normalization is much more important. If policy is trained without state normalization or without the scale factors λ_1 and λ_2 in the state normalization, there is no advantage in the training, i.e., the training is ineffective. The reason can be explained as below. Without the scale factors or state normalization, the values of b_t and $E_{H,t}$ are too large. With such large values, the randomly initialized neural networks will tend to output large values and have difficulty in learning for better results with large learning rates. Thus, the policy will become the greedy policy without the aid of our proposed state normalization.

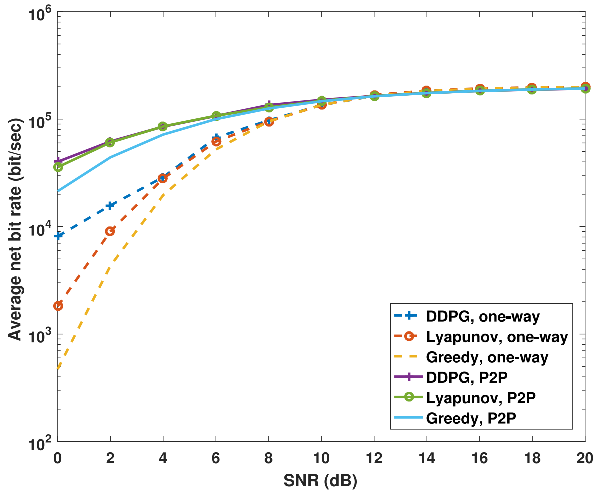


Fig. 6. Average net bit rate under the QPSK modulation.

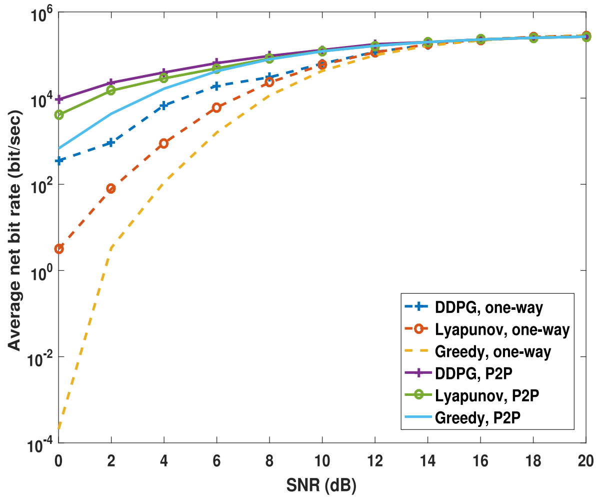


Fig. 7. Average net bit rate under the 8PSK modulation.

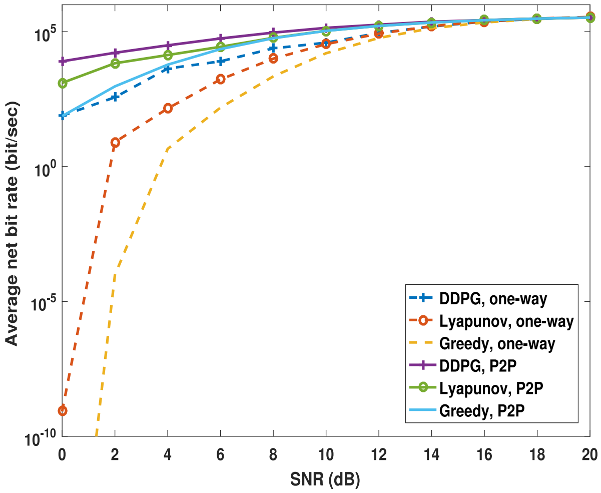


Fig. 8. Average net bit rate under the 16-QAM modulation.

D. Performance Comparison

We first compare the average net bit rate performance among the proposed DDPG method, the Lyapunov optimization [16], and the Greedy algorithm, under different modulations. The results are shown in Figs. 6–8. For the one-way relay network, the SNR refers to the signal-to-noise ratio of the SD link, i.e., $(\Psi_s \zeta_{sd}/N_0)$ and the channel noise power of the SD link and RD link is the same. For the point-to-point network, since there is no relay, the average channel-to-noise ratio is set to be SNR-10.

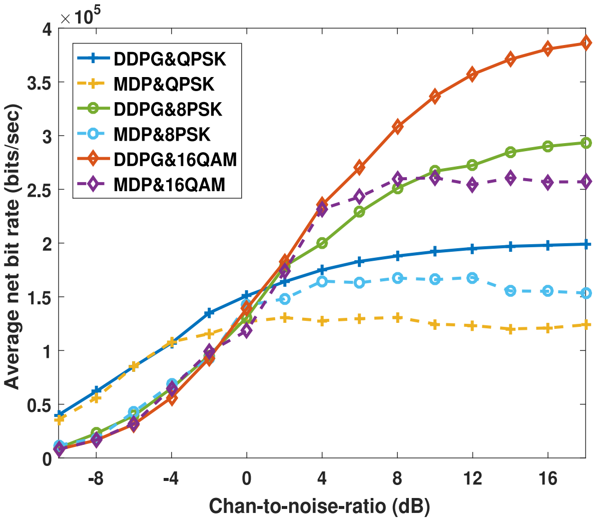


Fig. 9. Comparison between our method and MDP [8].

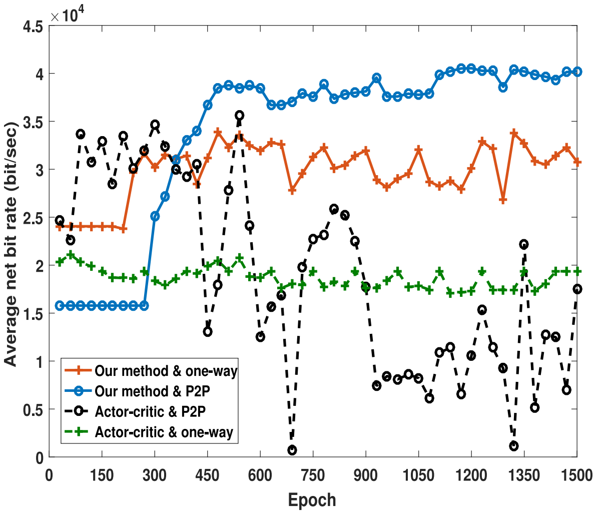


Fig. 10. Comparison between our method and actor–critic RL.

Fig. 6 shows the average net bit rate with the QPSK modulation under different methods. From the figure, we can see that the proposed method performs the best in both the point-to-point network and the one-way relay network, while the Lyapunov optimization performs better than the greedy algorithm. At the low SNR region, the performance gap among different methods is significant. When the SNR is 0 dB, the achieved average net bit rate with the proposed method is around 8×10^3, which raises approximately four times compared with the Lyapunov optimization, and ten times compared with the greedy strategy. For the point-to-point network, the gap is much smaller. This is mainly because there is no link with constant power supply in the point-to-point network, due to which the received SNR will be affected significantly. With the growth of SNR, the average net bit rate of different methods all saturates at the value of 2×10^5, which corresponds to the case with zero BER in (8).

Figs. 7 and 8 show the performance comparison of different policies with the 8PSK and 16-QAM modulations. Similar to the results in Fig. 6, the performance of the proposed method is the best, and the Lyapunov optimization performs better than the greedy algorithm, especially at the low SNR region. With the increase of the SNR, the net bit rate of all schemes saturates at the value of $(\chi_m L_S/T_P)$, i.e., 3×10^5 and 4×10^5. However, compared with the results in Fig. 6, we can see that the gaps among different strategies in Figs. 7 and 8 are much larger. The reason is that with the same energy consumed for transmission, the probability of successfully transmitting an entire packet under 16 QAM and 8PSK will be much smaller than that under QPSK. With the proposed method, the energy management is much more careful, which thus leads to much better performance.

Fig. 9 illustrates the performance comparison of the proposed method and the MDP algorithm in the point-to-point network. We do not evaluate the MDP in the one-way relay network since (14) in [8] cannot be derived with two channel links. The interval number of the solar, battery, channel, and action in MDP are set as 4, 8, 4, and 8, respectively. From the figure, we can see that at the low SNR region, the proposed method has a similar performance with the MDP algorithm, regardless the modulation. However, with the growth of the SNR, the performance of MDP finally converges at a much smaller value than that of the proposed method. This phenomenon may be due to the following reason. In MDP, if the average harvesting rate of the solar power is smaller than the basic action level, the transmitter has to wait several epoches for a basic energy quantum for transmission. As a result, in some management periods the transmitter are not capable of transmitting any bits because the energy amount in battery cannot reach the basic action power [8].

Finally, we compare the proposed method with the basic actor–critic method [30]. The structures of the actor net and the critic net in the basic actor–critic method are the same as the proposed method. Moreover, the learning rates and other setup parameters are the same as those in our methods, and the normalization has also been applied. The main difference is that in the basic actor–critic method there is no replay buffer and the action is estimated by a distribution instead of a deterministic function. From the result in Fig. 10, we can see that the basic actor–critic method is not able to learn an efficient state-action strategy in the stochastic energy harvesting problems. The energy management strategy cannot converge at a good value in the point-to-point network, and in one-way relay network the strategy cannot make any improvement with the actor–critic learning. Therefore, the basic actor–critic method cannot learn the complex state-action pattern in the stochastic energy harvesting wireless communications.

VI. Conclusion

In this paper, we studied the energy management problem in energy harvesting wireless networks. Our objective was to maximize the long-term average net bit rate in the point-to-point network and one-way relay network. We employed DDPG to train an optimal energy management policy to optimize the net bit rate. We introduced the training procedure of DDPG and derived the corresponding training and validating algorithms. We proposed a state normalization algorithm to make the training much faster and easier. We also theoretically analyzed the time and space complexity of the proposed training and validating algorithms. We conducted the parametric analysis through simulations to figure out the influence of the DDPG parameters, including the learning rate, the discount factor in Q-table, the length of the training sample, and state normalization. Compared with the state-of-the-art algorithms, the proposed algorithm achieves better performance in terms of long-term average net bit rate.

REFERENCES

[1] W. Sun, J. Liu, and H. Zhang, "When smart wearables meet intelligent vehicles: Challenges and future directions," *IEEE Wireless Commun.*, vol. 24, no. 3, pp. 443–461, Jun. 2017.

[2] M.-L. Ku, W. Li, Y. Chen, and K. J. R. Liu, "Advances in energy harvesting communications: Past, present, and future challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1384–1412, 2nd Quart., 2016.

[3] S. Sudevalayam and P. Kulkarni, "Energy harvesting sensor nodes: Survey and implications," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 3, pp. 443–461, 3rd Quart., 2011.

[4] S. Ulukus *et al.*, "Energy harvesting wireless communications: A review of recent advances," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 3, pp. 360–381, Mar. 2015.

[5] O. Ozel, K. Tutuncuoglu, J. Yang, S. Ulukus, and A. Yener, "Transmission with energy harvesting nodes in fading wireless channels: Optimal policies," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 8, pp. 1732–1743, Sep. 2011.

[6] Z. Wang, V. Aggarwal, and X. D. Wang, "Iterative dynamic water-filling for fading multiple-access channels with energy harvesting," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 3, pp. 382–395, Mar. 2015.

[7] R. Zhang, A. Nayak, S. Zhang, and J. Yu, "Energy-efficient sleep scheduling in WBANs: From the perspective of minimum dominating set," *IEEE Internet Things J.*, to be published.

[8] M.-L. Ku, Y. Chen, and K. J. R. Liu, "Data-driven stochastic models and policies for energy harvesting sensor communications," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 8, pp. 1505–1520, Aug. 2015.

[9] M.-L. Ku, W. Li, Y. Chen, and K. J. R. Liu, "On energy harvesting gain and diversity analysis in cooperative communications," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2641–2657, Dec. 2015.

[10] W. Li, M.-L. Ku, Y. Chen, and K. J. R. Liu, "On outage probability for two-way relay networks with stochastic energy harvesting," *IEEE Trans. Commun.*, vol. 64, no. 5, pp. 1901–1915, May 2016.

[11] W. Li, M.-L. Ku, Y. Chen, and K. J. R. Liu, "On outage probability for stochastic energy harvesting communications in fading channels," *IEEE Signal Process. Lett.*, vol. 22, no. 11, pp. 1893–1897, Nov. 2015.

[12] W. Li, M.-L. Ku, Y. Chen, K. J. R. Liu, and S. H. Zhu, "Performance analysis for two-way network-coded dual-relay networks with stochastic energy harvesting," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 5747–5761, Sep. 2017.

[13] A. Yadav, M. Goonewardena, W. Ajib, O. A. Dobre, and H. Elbiaze, "Energy management for energy harvesting wireless sensors with adaptive retransmission," *IEEE Trans. Commun.*, vol. 65, no. 12, pp. 5487–5498, Dec. 2017.

[14] J. Gong, Z. Y. Zhou, and S. Zhou, "On the time scales of energy arrival and channel fading in energy harvesting communications," *IEEE Trans. Green Commun. Netw.*, vol. 2, no. 2, pp. 482–492, Jun. 2018.

[15] Z. W. Ni and M. Motani, "Online policies for energy harvesting receivers with time-switching architectures," *IEEE Trans. Wireless Commun.*, vol. 18, no. 2, pp. 1233–1246, Feb. 2019.

[16] C. Qiu, Y. Hu, Y. Chen, and B. Zeng, "Lyapunov optimization for energy harvesting wireless sensor communications," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1947–1956, Jun. 2018.

[17] C. Qiu, Y. Hu, and Y. Chen, "Lyapunov optimized cooperative communications with stochastic energy harvesting relay," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1323–1333, Apr. 2018.

[18] Y. Hu, C. Qiu, and Y. Chen, "Lyapunov-optimized two-way relay networks with stochastic energy harvesting," *IEEE Trans. Wireless Commun.*, vol. 17, no. 9, pp. 6280–6292, Jul. 2018.

[19] F. Amirnavaei and M. Dong, "Online power control optimization for wireless transmission with energy harvesting and storage," *IEEE Trans. Wireless Commun.*, vol. 15, no. 7, pp. 4888–4901, Jul. 2016.

[20] Y. Cui, V. K. N. Lau, R. Wang, H. Huang, and S. Q. Zhang, "A survey on delay-aware resource control for wireless systems—Large deviation theory, stochastic Lyapunov drift, and distributed stochastic learning," *IEEE Trans. Inf. Theory*, vol. 58, no. 3, pp. 1677–1701, Mar. 2012.

[21] B. Cao, L. Zhang, Y. Li, D. Feng, and W. Cao, "Intelligent offloading in multi-access edge computing: A state-of-the-art review and framework," *IEEE Commun. Mag.*, vol. 57, no. 3, pp. 56–62, Mar. 2019.

[22] P. Blasco, D. Gündüz, and M. Dohler, "A learning theoretic approach to energy harvesting communication system optimization," *IEEE Trans. Wireless Commun.*, vol. 12, no. 4, pp. 1872–1972, Nov. 2013.

[23] R. F. Atallah, C. M. Assi, and J. Y. Yu, "A reinforcement learning technique for optimizing downlink scheduling in an energy-limited vehicular network," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 4592–4601, Jun. 2017.

[24] A. Ortiz, H. Al-Shatri, X. Li, T. Weber, and A. Klein, "Reinforcement learning for energy harvesting decode-and-forward two-hop communications," *IEEE Trans. Green Commun. Netw.*, vol. 1, no. 3, pp. 309–319, Sep. 2017.

[25] F. A. Aoudia, M. Gautier, and O. Berder, "RLMan: An energy manager based on reinforcement learning for energy harvesting wireless sensor networks," *IEEE Trans. Green Commun. Netw.*, vol. 2, no. 2, pp. 408–417, Jun. 2018.

[26] Y. F. Wei, F. R. Yu, M. Song, and Z. Han, "User scheduling and resource allocation in HetNets with hybrid energy supply: An actor-critic reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 17, no. 1, pp. 680–692, Jan. 2018.

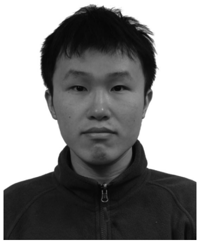
[27] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Feb. 2016.

[28] K. Cho and D. Yoon, "On the general BER expression of one- and two-dimensional amplitude modulations," *IEEE Trans. Commun.*, vol. 50, no. 7, pp. 1074–1080, Jul. 2002.

[29] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint, arXiv:1502.03167*, 2015.

[30] R. S. Suttion and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2014.

[31] *Cooperative Networks for Renewable Resource Measurements (CONFRRM) Solar Energy Resource Data*, NREL, Golden, CO, USA, Sep. 2012. [Online]. Available: http://www.nrel.gov/rredc

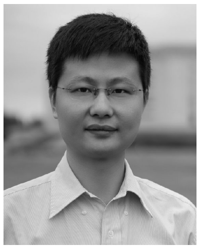


Chengrun Qiu received the B.S. degree in electronic information engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2018, where he is currently pursuing the master's degree with the School of Information and Communication Engineering.

His current research interests include reinforcement learning, stochastic optimization, and green communications.

Yang Hu received the B.S. and Ph.D. degrees in electrical engineering from the University of Science and Technology of China, Hefei, China, in 2004 and 2009, respectively.

She was with the University of Maryland Institute for Advanced Computer Studies as a Research Associate from 2010 to 2015. She is currently an Associate Researcher with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China. Her current research interests include computer vision, machine learning, and multimedia signal processing.



Yan Chen (SM'14) received the bachelor's degree from the University of Science and Technology of China, Hefei, China, in 2004, the M.Phil. degree from the Hong Kong University of Science and Technology, Hong Kong, in 2007, and the Ph.D. degree from the University of Maryland at College Park, College Park, MD, USA, in 2011.

He was with Origin Wireless, Inc., Greenbelt, MD, USA, as a Founding Principal Technologist. Since 2015, he has been a Full Professor with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China. His current research interests include multimedia, signal processing, game theory, and wireless communications.

Dr. Chen was a recipient of multiple honors and awards, including the Best Student Paper Award at the PCM in 2017, the Best Student Paper Award at the IEEE ICASSP in 2016, the Best Paper Award at the IEEE GLOBECOM in 2013, the Future Faculty Fellowship and Distinguished Dissertation Fellowship Honorable Mention from the Department of Electrical and Computer Engineering in 2010 and 2011, the Finalist of the Dean's Doctoral Research Award from the A. James Clark School of Engineering, the University of Maryland in 2011, and the Chinese Government Award for outstanding students abroad in 2010.

Bing Zeng (M'91–SM'13–F'16) received the B.Eng. and M.Eng. degrees in electronic engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 1983 and 1986, respectively, and the Ph.D. degree in electrical engineering from the Tampere University of Technology, Tampere, Finland, in 1991.

He was a Post-Doctoral Fellow with the University of Toronto, Toronto, ON, Canada, from 1991 to 1992 and a Researcher with Concordia University, Montreal, QC, Canada, from 1992 to 1993. He then joined the Hong Kong University of Science and Technology (HKUST), Hong Kong. After 20 years of service at HKUST, he returned to UESTC in 2013, through China's 1000-Talent-Scheme. At UESTC, he leads the Institute of Image Processing to work on image and video processing, 3-D and multiview video technology, and visual big data. During his tenure at HKUST and UESTC, he graduated over 30 master and Ph.D. students, received about 20 research grants, filed 8 international patents, and published over 260 papers. Three representing works are as follows: one paper on fast block motion estimation, published in the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (TCSVT) in 1994, has so far been SCI-cited over 1000 times (Google-cited over 2200 times) and currently stands at the 8th position among all papers published in this transactions; one paper on smart padding for arbitrarily shaped image blocks, published in IEEE TCSVT in 2001, leads to a patent that has been successfully licensed to companies; and one paper on directional discrete cosine transform, published in IEEE TCSVT in 2008.

Dr. Zeng was a recipient of the 2011 IEEE CSVT Transactions Best Paper Award, the Best Paper Award at ChinaCom three times (2009 Xi'an, 2010 Beijing, and 2012 Kunming), and the Second Class Natural Science Award (the first recipient) from Chinese Ministry of Education in 2014. He served as an Associate Editor for the IEEE TCSVT for eight years and received the Best Associate Editor Award in 2011. He was the General Co-Chair of VCIP-2016 and PCM-2017. He was elected as an IEEE Fellow in 2016 for contributions to image and video coding.