Spectrum-Agile Cognitive Interference Avoidance Through Deep Reinforcement Learning

Mohamed A. Aref and Sudharman K. Jayaweera(✉)

Communications and Information Sciences Laboratory (CISL), ECE Department, University of New Mexico, Albuquerque, NM, USA
{maref,jayaweera}@unm.edu

Abstract. This work introduces a spectrum-agile wideband autonomous cognitive radio (WACR) that is capable of avoiding interference and jamming signals. Proposed cognitive technique is based on deep reinforcement learning (DRL) that uses a double deep Q-network (DDQN). Moreover, it introduces new definitions for the state and the operation parameters that enable the WACR to collect information about the RF spectrum of interest in both time and frequency domains. The simulation results show that the proposed technique can efficiently learn an effective strategy to avoid harmful signals in a wideband partially observable environment. Furthermore, the experiments on an over-the-air channel inside a laboratory show that the proposed algorithm can rapidly adapt to sudden changes in the surrounding RF environment making it suitable for real-time applications.

Keywords: Deep Q-network · Deep reinforcement learning · Interference avoidance · Wideband autonomous cognitive radios

1 Introduction

With its ability to automatically extract important features from data, deep learning (DL) has made major breakthroughs in many applications such as computer vision, natural language processing, medical diagnosis, image and speech recognition [1–3]. In recent years, this has prompted researchers to investigate application of DL techniques in the wireless communications domain. The RF spectrum domain, however, has different characteristics compared with other domains including high data rates, representation of RF waveforms as complex numbers and time-varying multipath wireless channels. These all make the task of applying DL in the RF spectrum domain challenging because it requires modifications to existing DL algorithms or develop new ones. In the coming years, the DL is expected to play an important role in future wireless communications networks design including Internet of things (IoT), Unmanned Aerial Vehicles (UAVs) and the 6th generation (6G) cellular communication systems.

A. Kliks et al. (Eds.): CrownCom 2019, LNICST 291, pp. 218–231, 2019.
https://doi.org/10.1007/978-3-030-25748-4_17

Recently, the wideband autonomous cognitive radios (WACRs) have been proposed as an emerging technology to achieve spectrum situational awareness and signal intelligence [4–6]. With its ability to sense, learn and take decisions, a WACR may be a good candidate to apply DL techniques and especially deep reinforcement learning (DRL) to effectively address challenges that may be difficult to solve with the traditional machine learning techniques. The DRL is one of the widely used DL techniques in applications that require autonomous decision-making [7–9]. The DRL explores the advantage of deep neural networks to improve the training and the learning process of the traditional reinforcement learning making it suitable for systems with a large state-action space [7,9,10]. Most existing DRL techniques are based on deep Q-network (DQN) algorithm that extends the Q-leaning by using a convolutional neural network (CNN) instead of the Q-table to learn an approximate Q-function [7,10].

The DRL has previously been proposed for several applications in cognitive radio networks (CRNs) including power control, network access and connectivity preservation [9,11–16]. Another important application is the network security in which the CR adopts DRL to avoid jamming and other malicious attacks. One of the first works that uses DQN for the anti-jamming in CRN can be found in [14]. The system model in [14] assumes one secondary user (SU), one primary user (PU) and two jammers. The SU adopts a DQN with CNN to learn an efficient frequency hopping policy and decide whether to leave the area of heavy jamming and connect to another base station. One of the drawbacks of the proposed approach in [14] is that the state definition is based on the signal-to-interference-plus-noise ratio (SINR) estimates of the signals. In practice, SINR may take arbitrary value and the SINR estimates may not be perfect.

The authors in [15] extend the model in [14] by adding mobility features to the receiver allowing it to change its location. Using the same state and utility definitions in [14], the receiver is considered an agent that needs to learn an optimal policy using the DQN. However, the mobility capabilities may not be available for the SU and its corresponding receiver in many real-time applications. In [16], the authors considered the same problem formulation as in [14] in which the SU attempts to learn an optimal frequency hopping strategy. The authors in [16] used the spectrum vector as their system state that contains the received power spectral density (PSD) function at different time instants. This framework, however, is not applicable for wideband applications where the agent cannot sense all frequency channels simultaneously.

The goal of this paper is to design a spectrum-agile WACR that is capable of finding spectrum opportunities in a heterogeneous RF environment contested by jamming and crowded with interference signals. We propose a cognitive interference and jamming resilience technique that is suitable for real-time applications and mitigates limitations in the above mentioned previous work. Our proposed technique is based on double deep Q-network (DDQN) algorithm [17]. The advantages of the proposed approach can be summarized as follows:

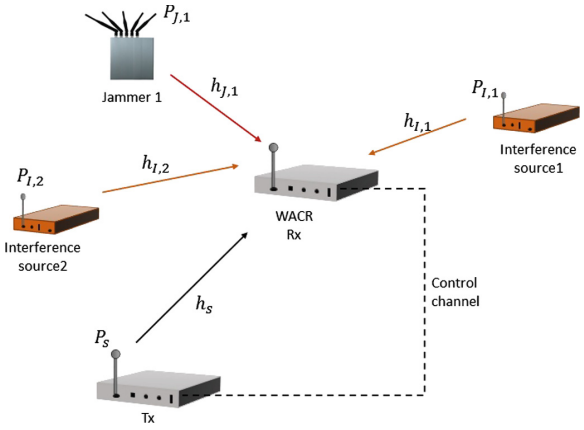


Fig. 1. System model.

- Ability to work in a partially observable wideband spectrum environment making it suitable for existing hardware, including the ones with limited instantaneous bandwidth.
- New simple definitions for the state and operation parameters that can represent more information about the surrounding RF environment in both time and frequency domains.
- A fast learning algorithm that can rapidly reconfigure to tackle sudden changes in the RF environment making it suitable for real-time applications in heterogeneous environments.

The rest of this paper is organized as follows: the system model is introduced in Sect. 2. Next, the proposed DDQN algorithm is discussed in details in Sect. 3. The performance evaluation is shown in Sect. 4 including both simulation and experimental results in an over-the-air channel inside a laboratory. Finally, Sect. 5 contains the concluding remarks.

2 System Model

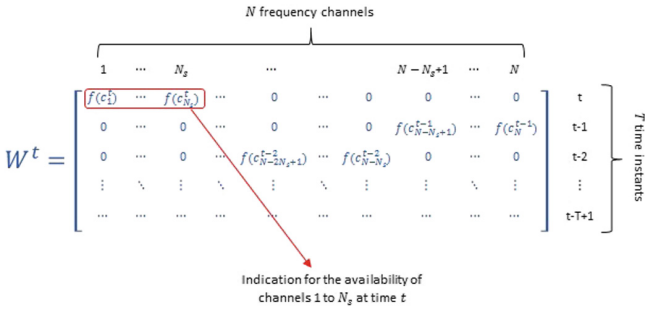
Let us consider a WACR that is operating in a heterogeneous RF environment that includes multiple interference and jamming signals as shown in Fig. 1. The WACR is considered as the receiver in the communications link of interest, while the transmitter device may or may not have cognitive capabilities. The objective of the WACR is to choose a frequency channel with highest SINR for communications at every time instant. It is assumed that the frequency synchronization between the receiver and the transmitter is done through a secured common control channel as shown in Fig. 1. A centralized controller (e.g. a base station) or frequency rendezvous algorithms could be used as alternatives for the common control channel to maintain the frequency synchronization between the two nodes [18,19].

The RF spectrum of interest is assumed to have N possible channels. At time t, the WACR chooses an action, denoted by $a^t \in \{1, \cdots, N\}$, that represents the index of the channel for communications at time $t+1$. The transmitter sends the signal of interest with a given power P_s. The channel power gain from the transmitter to the WACR is given by h_s. The interference source i and the jammer j send their signals with given powers $P_{I,i}$ and $P_{J,j}$, while their channel power gains to the WACR are $h_{I,i}$ and $h_{J,j}$, respectively. The received SINR of the WACR at channel n and time t can be expressed as

$$\mu_n^t = \frac{h_s P_s}{\sigma^2 + \sum_i h_{I,i} P_{I,i} + \sum_j h_{J,j} P_{J,j}} \ , \tag{1}$$

where σ^2 is the receiver noise power, assuming additive white Gaussian noise.

Due to hardware constraints, the WACR may not be able to sense all the N channels simultaneously. Assume that at any time instant the WACR can sense only N_s channels, with $N_s \leq N$. At time t, the WACR can estimate the power spectral density c_n^t for the sensed channel n. The WACR can then identify the availability of channel by comparing c_n^t with an appropriate threshold c_{th} that is designed based on noise floor estimation [4]. Let $f(c_n^t) = 1$ denotes the unavailability of the channel for $c_n^t > c_{th}$, otherwise $f(c_n^t) = 0$. At any given time, sensing is assumed to be performed on a different channel than the one used for communications. Thus, the WACR can sense the surrounding RF spectrum while maintaining the communications link. The sensing process can adopt any strategy (e.g. sweeping or random selections) based on the application of interest. In the following, we will assume that the WACR adheres to sweeping sensing strategy that sweeps sequentially over the spectrum of interest. Then, at time t, a sensing matrix W^t that stores the sensing results of all channels for T successive time instants up to time t is defined as follows:



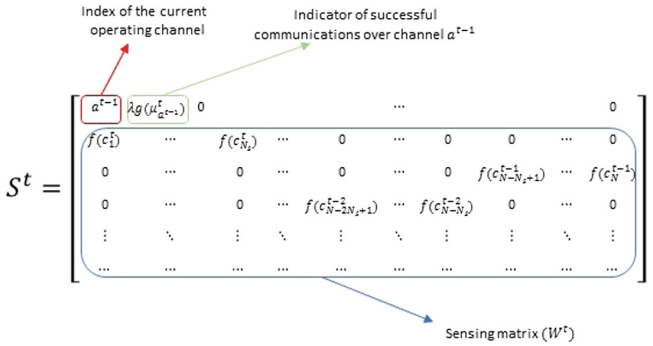
The columns of W^t represent the different channels, while the rows represent the temporal memory depth. For each row, there are N_s values that indicate the availability of the sensed channels at the corresponding time instant. If $N_s \neq N$, remaining entries in each row are filled with zeros. Since the sensing matrix contains rich information about the RF environment in both frequency and time domains, it is used as a part of the state. In addition, the state definition also includes the index of the current channel used for communications in addition

to an indication whether the communications over this channel is successful or not.

Let μ_{th} denote the required SINR threshold for successful communications. Then, an indicator function for the communications over channel n at time t can be defined as follows:

$$g(\mu_n^t) = \begin{cases} 1 & \text{if } \mu_n^t > \mu_{th} \text{ (success)} \\ 0 & \text{if } \mu_n^t \leq \mu_{th} \text{ (failure)} \end{cases} \tag{2}$$

The state at time t is then represented by a $(T+1) \times N$ matrix as shown below:



where $\lambda > 1$ is a weighting factor that may be optimized to achieve efficient learning. For sufficiently large T, the state may include information about all the channels of interest, ordered in time. Since there is only two possible values: 0 and 1 (denoting availability and unavailability, respectively), for each channel, the proposed state definition is less complicated compared with previous definitions that include SINR estimates as in [14] or received PSD as in [16].

The interference avoidance problem can be modeled as a Markov decision process (MDP) [20]. By choosing action a^t at time t, the WACR moves from its current state S^t to a new state S^{t+1} and receives a reward. The reward of choosing channel a^t for transmission while in state S^t is defined as the received SINR value $r(S^t, a^t) = \mu_{a^t}^{t+1}$. Note that, the reward value of state S^t and action a^t is obtained in the next time instant $t+1$.

3 Proposed Double Deep *Q*-Network (DDQN) Algorithm

Reinforcement learning (RL) has shown to be a good candidate for learning in MDP environments [10]. It is based on delayed-reward principle in which the agent receives a reward from the environment after executing each action [4]. The value of the reward indicates how good or bad the action is. The objective of the agent is then to choose actions that maximize the rewards. In our scenario, the WACR attempts to learn a channel selection policy that maximizes the received SINR at each time instant.

The traditional RL approaches such as Q-learning, however, may not be the best technique in our scenario for several reasons. First, we are dealing

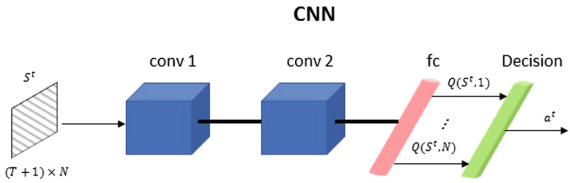


Fig. 2. CNN network structure of the proposed DDQN-based interference avoidance technique.

with a two-dimensional state. Second, the number of possible states can become extremely large even with few channels and a short memory depth. Furthermore, the rate of convergence of Q-learning may not be sufficient for real-time applications because it needs long time to explore and gain knowledge of the entire system. Hence, in this paper we propose using DDQN algorithm, an extension of the DQN that is developed by Google DeepMind team [17].

The basic idea of the DQN is to combine reinforcement learning with deep neural networks, more specifically, a CNN [7]. For each time t, the previously defined state S^t is used as an input to the proposed CNN. Then, the CNN attempts to estimate the Q-value $Q(S^t, a^t)$ for each possible action $a^t \in \{1, \cdots, N\}$. Several tests were performed to determine the best CNN design and the configuration of each layer to achieve consistently high performance while keeping the structure as simple as possible. Figure 2 shows the network structure of the proposed CNN which consists of 2 convolutional layers and 1 fully connected layer. The first convolutional layer (conv1) includes 10 filters with size 1×1 and stride 1. The second convolutional layer (conv2) has 20 filters of size 2×2 and stride 1. Both convolutional layers use rectified linear unit (ReLU) as the activation function. The fully connected layer (fc), on the other hand, has N rectified linear units that are used to output the Q-value estimates for each possible action. Finally, the WACR decides the action a^t corresponding to the maximum Q-value estimate.

For training, the DQN uses experience replay in which we store WACR's experiences $x^t = (S^t, a^t, \mu_{a^t}^{t+1}, S^{t+1})$ at each time t in a data set $\mathcal{D}^t = \{x^1, \cdots, x^t\}$. Let θ^t represents the weights of the proposed Q-network (CNN) at time t. During learning at time t, we draw an experience $x^k \sim U(\mathcal{D}^t)$, where U denotes the uniform distribution on $\mathcal{D}^t$ with $1 \leq k \leq t$, from the set of the stored experiences. The network parameters θ^t are then updated according to a stochastic gradient descent algorithm using the following loss function [7]:

$$L(\theta^t) = \mathbb{E}_{(S^t, a^t, \mu_{a^t}^{t+1}, S^{t+1}) \sim U(D)}[(\eta - Q(S^t, a^t; \theta^t)^2] \tag{3}$$

where η is the target optimal Q-value given by

$$\eta = \mu_{a^t}^{t+1} + \gamma \max_{a'} Q(S^{t+1}, a'; \hat{\theta}^t) \tag{4}$$

Algorithm 1. DDQN-aided proposed interference avoidance algorithm with experience replay

Initialize:
  Parameters λ, γ, ϵ, K
  The weights θ of the Q-network
  The weights $\hat{\theta}$ of the target Q-network
for each time t **do**
  Observe $\mu^t_{a^{t-1}}$, c^t_i, $\forall i \in \mathcal{C}^t$
  Obtain W^t and S^t
  With probability ϵ:
    Choose $a^t \in \{1, \cdots, N\}$ at random
  Otherwise:
    Obtain $Q(S^t, a')$ from the proposed CNN $\forall a'$
    Select $a^t = \arg\max_{a'} Q(S^t, a'; \theta^t)$
  Use channel a^t for communications at time $t+1$
  Store new experience $x^{t-1} = (S^{t-1}, a^{t-1}, \mu^t_{a^{t-1}}, S^t)$ in data set $\mathcal{D}$
  for k = 1, $\cdots$, K **do**
    Select $x^k = (S^k, a^k, \mu^{k+1}_{a^k}, S^{k+1}) \sim U(\mathcal{D})$
    Compute η from (5)
    Compute the gradient of the loss function (3)
    Update θ^t
  end for
  Reset $\hat{\theta}^t = \theta^t$ for every fixed number of iterations.
end for

with $\hat{\theta}^t$ representing the weights of the target Q-network. This process can be repeated for K times at each time t in which θ^t is updated according to K randomly selected experiences.

The max operator in (4) uses the same value $Q(S^{t+1}, a'; \hat{\theta}^t)$ to decide which action is the best and to evaluate the optimal Q-value which might produce overestimated values degrading the learning process and the convergence rate [17, 21]. In order to overcome this problem, we use DDQN to decouple the selection and the evaluation operations. In this case, the original Q-network (with weights θ^t) is used for action selection and the target Q-network (with weights $\hat{\theta}^t$) is used to estimate the Q-value associated with the selected action. Thus, the target value η of (4) can be rewritten as follows:

$$\eta = \mu^{t+1}_{a^t} + \gamma\, Q(S^{t+1}, \arg\max_{a'} Q(S^{t+1}, a'; \theta^t); \hat{\theta}^t) \tag{5}$$

Algorithm 1 summarizes the proposed DDQN-based interference avoidance approach. For each time t, the WACR computes the received SINR $\mu^t_{a^{t-1}}$ on the current channel a^{t-1}. Let $\mathcal{C}^t$ represent the set of N_s channel indices that the WACR is sensing at time t. The WACR identifies the power spectral density c^t_i at each channel $i \in \mathcal{C}^t$ and updates the sensing matrix W^t. With the knowledge of a^{t-1}, $\mu^t_{a^{t-1}}$ and W^t, the WACR can obtain the current state S^t. The DDQN algorithm takes the state S^t as an input and estimates the Q-values for all

possible actions. The optimal action $a^t = \arg\max_{a'} Q(S^t, a'; \theta^t)$ is chosen with a high probability $1-\epsilon$, and a random action $a^t \in \{1, \cdots, N\}$ is selected uniformly with low probability ϵ to avoid staying in a local optima.

4 Performance Evaluation and Experimental Results

4.1 Simulation Results

Simulations have been performed to evaluate the performance of our proposed interference avoidance technique. The following parameters are used: $N = 6$, $T = 5$, $N_s = 2$, $K = 5$, $\epsilon = 0.1$, $\gamma = 0.4$, $\lambda = 10$, $\sigma^2 = 1$, $c_{th} = 2$, $\mu_{th} = 2$ and learning rate of 0.1. With these parameter values, state S^t at any time t is a 6×6 matrix which is the input to the CNN. Jamming signal j is transmitted with power $P_{J,j} = 8\,\text{mW}$ with a channel power gain to the WACR $h_{J,j} = 0.7$. On the other hand, any interference signal i has a transmit power of $P_{I,i}$ that can take any value between 3 mW and 6 mW, while the channel power gain to the WACR $h_{I,i}$ is ranging from 0.4 to 0.9. For each interference source i, the values of $P_{I,i}$ and $h_{I,i}$ are chosen randomly from the predefined sets. Our signal of interest is transmitted with power $P_s = 5\,\text{mW}$ and the channel power gain to the WACR is $h_s = 0.8$. Hence, the optimal SINR value at any channel is 4 which corresponds to WACR selecting a channel free of interference and jamming.

As a benchmark, we used DQN, Q-learning and random channel selection techniques to evaluate our proposed DDQN technique [5]. Similar to the DDQN, the action and the reward of the DQN and Q-learning at time t are the index of the channel $a^t \in \{1, \cdots, N\}$ and the received SINR value μ^t, respectively. The DQN uses the same state definition S^t as in the proposed algorithm. The Q-learning, however, uses a simplified version of the original proposed state that does not include the sensing matrix W^t. Instead, the state of the Q-learning algorithm at time t is represented by $S_Q^t = [a_Q^{t-1}, \lambda g(\mu^t_{a_Q^{t-1}})]$ so that the number of possible states is $2N$. On the other hand, in the random technique, the WACR randomly chooses a channel for communications.

Three test cases are considered with different interference and jamming signal scenarios. Table 1 shows the performance comparison with a scenario description for each test case. Test case 1 represents a simplified scenario in which there are

Table 1. Performance comparison: normalized accumulated reward values after 10,000 iterations.

Test case	Scenario	Proposed	DQN	Q-learning	Random	Optimal
1	2 interference signals	3.73	3.68	3.62	3.02	4
2	3 interference signals	3.65	3.56	3.52	2.57	4
3	3 interference signals and Markov jammer	3.12	3.07	2.84	2.14	4

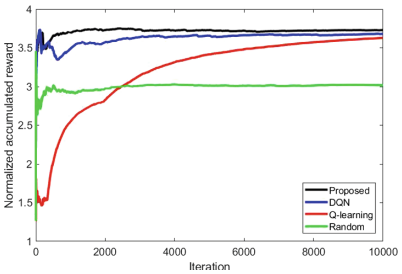


Fig. 3. $T = 5$: normalized accumulated reward (SINR) for test case 1.

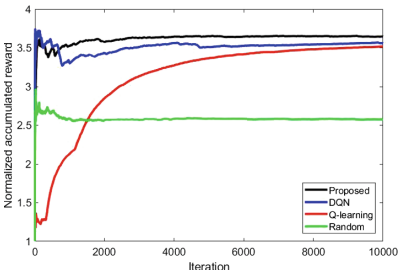


Fig. 4. $T = 5$: normalized accumulated reward (SINR) for test case 2.

only two interference sources that transmit continuously their signals over two dedicated channels. Figure 3 shows the normalized accumulated reward for this scenario. Two main observations can be obtained from Fig. 3: (1) The proposed DDQN technique achieves a higher SINR than DQN, Q-learning and random techniques. (2) The proposed DDQN technique has a faster convergence than both the DQN and Q-learning.

In test case 2, an extra interference source is added on a third dedicated channel besides the two interference sources described above. This source, however, does not operate continuously. Instead, it switches between ON and OFF in a random manner. From Fig. 4, we may observe that the proposed DDQN technique outperforms both Q-learning and random techniques while having a similar performance to the DQN.

In test case 3, there is a Markov jammer operating besides the 3 interference signals described in test case 2. The Markov jammer selects a channel to jam based on a Markov chain as shown in Fig. 5 where $p_h = 0.8$ and $p_l = 0.2$. Figure 6 shows the normalized accumulated reward for this scenario: (a) for 10,000 iterations (b) for 2,000 iterations to have a closer look on the convergence rate. Again, from Fig. 6, the proposed DDQN technique shows better performance in terms of SINR and convergence rate compared to those achieved with the DQN and Q-learning.

Figure 7 shows the normalized accumulated reward for test case 3 for $T = 1$, $T = 5$ and $T = 10$. Part (a) of the figure shows the full iterations while part (b) only focuses on the beginning of the iterations to analysis the convergence rate. Note that, the state matrix dimensions at any time t in the case of $T = 1$ and $T = 10$ are 2×6 and 11×6, respectively. Figure 7 shows that reducing the temporal memory depth to $T = 1$ has a negative impact on the performance especially if the number of sensing channels is less than the total number of channels ($N_s = 2$ and $N = 6$).

On the other hand, both cases of $T = 10$ and $T = 5$ converge to the same accumulated reward value after 10,000 iterations as shown in Fig. 7 (a). This is because when $T = 5$, the state includes information about all the channels arranged in time from the newest to the oldest. Increasing the memory depth to

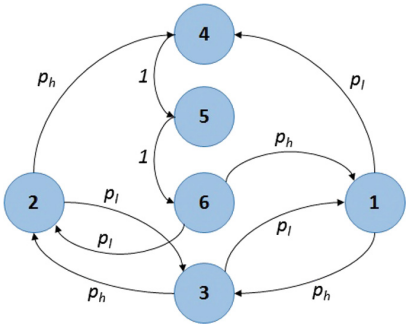


Fig. 5. Markov jammer selection strategy for test case 3 with 6 channels.

$T = 10$, will only add outdated information about the same channels. It does not seem to provide any significant new information since the most updated information about all the channels are already included with $T = 5$. However, this outdated information increases the state size that makes the computations more complex. These results show that choosing a suitable T value can be essential depending on the values of N and N_s.

4.2 Experimental Results

The experiments are performed inside the Communications and Information Sciences Laboratory (CISL) in the ECE Department at the University of New Mexico. The experiment setup consists of a USRP 2943R from National Instruments that is used as the WACR. The proposed cognitive interference-avoidance technique is implemented in LabVIEW on a DELL PRECISION TOWER 5810 PC with a built-in MATLAB interface to run the deep learning algorithm. The USRP interacts with the LabVIEW through a high speed PCIe connection.

The spectrum of interest is 240 MHz from 1.92 GHz to 2.16 GHz which is divided into 10 channels with 24 MHz each. The parameters used in the proposed DDQN are as follows: $N = 10$, $N_s = 1$, $T = 7$, $K = 5$, $\epsilon = 0.1$, $\gamma = 0.4$ and $\lambda = 10$. From spectrum observation, the noise floor threshold is set to -95 dBm. Any channel other than the one used by the WACR with received power above this threshold is considered unavailable. The USRP uses an IQ rate of $24 Msamples/sec$, acquisition time of 0.16ms and RX gain = 20 dB. Figure 8 shows the whole spectrum of interest as observed on the KEYSIGHT N9952A spectrum analyzer. It is clear from Fig. 8 that all but channel 5, 6 and 7 are occupied with different signals. Hence, if the proposed cognitive interference-avoidance algorithm works properly, the WACR has to choose a channel from these three channels.

The experiment consists of two stages. In the first stage we evaluate our proposed algorithm in the spectrum described above. We ran this stage for 300 iterations, in which each iteration represents a single sensing duration. The total time for this stage is about 489 s. The WACR adopts a random sensing strategy

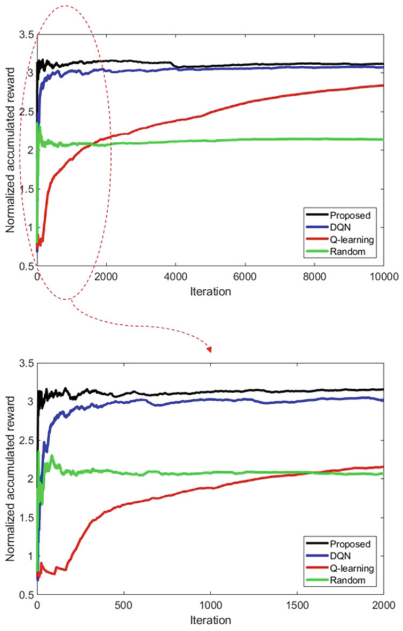


Fig. 6. $T = 5$: normalized accumulated reward (SINR) for test case 3.

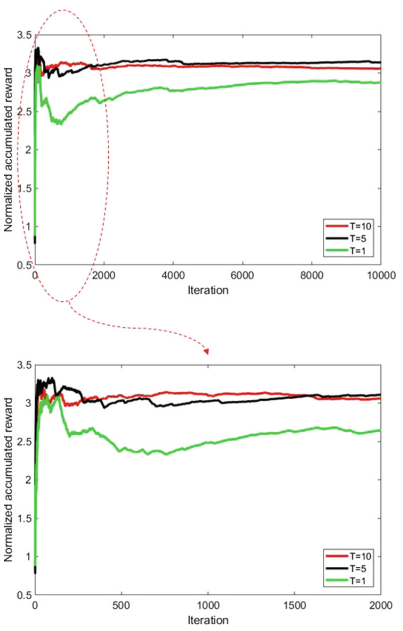


Fig. 7. Normalized accumulated reward (SINR) for test case 3 for different temporal memory depth values using the proposed algorithm.

in which it randomly selects a channel to sense for each iteration. Figure 9 shows the number of times that the WACR was able to avoid channels with interference as a percentage of the total number of iterations. Figure 10 shows whether the actions selected by the WACR correspond to a channel free of interference or not. From the figures, we can notice that the proposed DDQN algorithm was able to learn an optimal policy after a few number of iterations (approx. 40 iterations). In this experiment the WACR learned to operate in channel 6 which is free of interference.

An interesting question is how the WACR will react to sudden changes in the RF environment. A good learning algorithm should make the WACR adjust to this new condition rapidly. Thus, in the second stage of our experiment we generated an interference signal in channel 6 starting at the 301^{st} iteration. It can be observed from Fig. 10 that proposed DDQN algorithm reacts very fast and switch to a new interference-free location (channel 5).

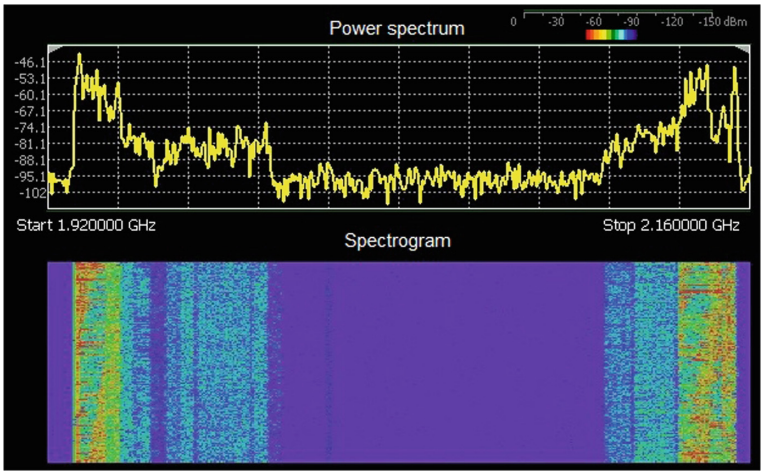


Fig. 8. Power spectrum and its corresponding spectrogram for start freq. = 1.92 GHz and stop freq. = 2.16 GHz.

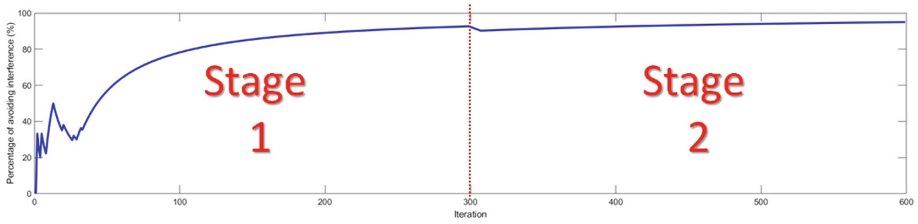


Fig. 9. The percentage of selecting interference-free channels.

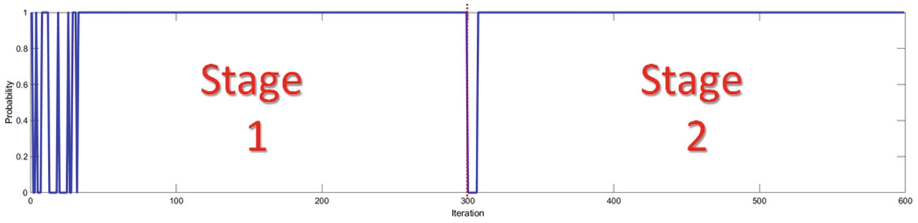


Fig. 10. Probability of selecting an interference-free channel for each iteration

5 Conclusion

In this paper, we have studied cognitive interference avoidance through spectrum agility. The proposed technique is based on DDQN algorithm with CNN. The WACR uses two separate channels for sensing and communications. The sensing operation is used to create the sensing matrix that includes information about the availability of different channels of interest. The sensing matrix along with

the chosen communications channel and an indication of the success/failure of the communications over this channel form the state of the DDQN. The proposed technique was evaluated through various test cases that include multiple interference and jamming signals. Both simulation and experimental results showed that the proposed algorithm is suitable for real-time applications and can operate over wideband spectrum. Furthermore, the proposed technique was shown to rapidly adapt to sudden changes in the surrounding RF environment.

Acknowledgment. This research was sponsored in part by the Army Research Laboratory and was accomplished under Grant Number W911NF-17-1-0035. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

1. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016). http://www.deeplearningbook.org
2. Deng, L.: A tutorial survey of architectures, algorithms, and applications for deep learning. APSIPA Trans. Signal Inf. Process. **3**, e2 (2014)
3. Litjens, G., et al.: Deep learning as a tool for increased accuracy and efficiency of histopathological diagnosis. Nat. Sci. Rep. **6**, 26286 (2016)
4. Jayaweera, S.K.: Signal Processing for Cognitive Radios, 1st edn. Wiley, New York (2014)
5. Aref, M.A., Jayaweera, S.K., Machuzak, S.: Multi-agent reinforcement learning based cognitive anti-jamming. In: IEEE Wireless Communications and Networking Conference (WCNC 17) , San Francisco, CA, March 2017
6. Jayaweera, S.K., Aref, M.A.: Cognitive engine design for spectrum situational awareness and signals intelligence. In: The 21st International Symposium On Wireless Personal Multimedia Communications (WPMC 18), Chiang Rai, Thailand (2018)
7. Mnih, V., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529 (2015)
8. Li, H., Wei, T., Ren, A., Zhu, Q., Wang, Y.: Deep reinforcement learning: framework, applications, and embedded implementations: invited paper. In: 2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Irvine, CA, USA, November 2017
9. Luong, N.C., et al.: Applications of deep reinforcement learning in communications and networking: a survey. arXiv, eprint arXiv:1810.07862 [cs.NI] (2018)
10. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
11. Huang, W., Wang, Y., Yi, X.: Deep q-learning to preserve connectivity in multi-robot systems. In: Proceedings of the 9th International Conference on Signal Processing Systems (ICSPS 2017) (2017)
12. Li, X., Fang, J., Cheng, W., Duan, H., Chen, Z., Li, H.: Intelligent power control for spectrum sharing in cognitive radios: a deep reinforcement learning approach. IEEE Access **6**, 25463–25473 (2018)

13. Liu, S., Hu, X., Wang, W.: Deep reinforcement learning based dynamic channel allocation algorithm in multibeam satellite systems. IEEE Access **6**, 15733–15742 (2018)
14. Han, G., Xiao, L., Poor, H.V.: Two-dimensional anti-jamming communication based on deep reinforcement learning. In: IEEE International Conference on Acoustics, Speech and Signal Processing, New Orleans, LA, USA, March 2017
15. Xiao, L., Jiang, D., Wan, X., Su, W., Tang, Y.: Anti-jamming underwater transmission with mobility and learning. IEEE Commun. Lett. **22**(3), 542–545 (2018)
16. Liu, X., Xu, Y., Jia, L., Wu, Q., Anpalagan, A.: Anti-jamming communications using spectrum waterfall: a deep reinforcement learning approach. IEEE Commun. Lett. **22**(5), 998–1001 (2018)
17. Van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double q-learning. In: The Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16), AZ, USA, Phoenix, February 2016
18. Theis, N.C., Thomas, R.W., DaSilva, L.A.: Rendezvous for cognitive radios. IEEE Trans. Mobile Comput. **10**(2), 216–227 (2011)
19. Pu, D., Wyglinski, A.M., McLernon, M.: An analysis of frequency rendezvous for decentralized dynamic spectrum access. IEEE Trans. Veh. Technol. **59**(4), 1652–1658 (2010)
20. Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley, Hoboken (2014)
21. Van Hasselt, H.: Double q-learning. In: Advances in Neural Information Processing systems 23 (NIPS 2010), pp. 2613–2621 (2010)