2014

Intelligent Maximum Power Extraction Control for Wind Energy Conversion Systems Based on Online Q-learning with Function Approximation

Chun Wei
University of Nebraska-Lincoln, cwei@huskers.unl.edu

Zhe Zhang
University of Nebraska-Lincoln, zhang.zhe@huskers.unl.edu

Wei Qiao
University of Nebraska–Lincoln, wqiao@engr.unl.edu

Liyan Qu
University of Nebraska-Lincoln, lqu2@unl.edu

Energy Conversion Congress and Exposition (ECCE), 2014 IEEE
Year: 2014
Pages: 4911 - 4916, DOI: 10.1109/ECCE.2014.6954074

Intelligent Maximum Power Extraction Control for Wind Energy Conversion Systems Based on Online Q-learning with Function Approximation

Chun Wei, Zhe Zhang, Wei Qiao, and Liyan Qu
Power and Energy Systems Laboratory
Department of Electrical Engineering
University of Nebraska-Lincoln
Lincoln, NE, 68588-0511 USA
cwei@huskers.unl.edu; zhang.zhe@huskers.unl.edu; wqiao@engr.unl.edu; lqu2@unl.edu

Abstract—**This paper proposes an intelligent maximum power point tracking (MPPT) algorithm for variable-speed wind energy conversion systems (WECSs) based on an online Q-learning algorithm. Instead of using the conventional Q-learning that uses a lookup table to store the action values for the discretized states, artificial neural networks (ANNs) are used as function approximators to output the action values by using the electrical power and rotor speed of the generator as inputs. This eliminates the need for a large storage memory. The proposed method learns the optimal speed control strategy of the WECS by updating the connecting weights of the ANNs, which has a lower computational cost than the conventional Q-learning method. Moreover, the knowledge of wind turbine characteristics or wind speed measurement is not required in the proposed method. The proposed method is validated by simulations for a WECS equipped with a doubly-fed induction generator (DFIG) and experimental results for an emulated WECS equipped with a permanent-magnet synchronous generator (PMSG).**

I. INTRODUCTION

Wind power is becoming an important alternative source for electricity supply. By the middle of 2013, the worldwide wind power capacity has reached 296 GW, which can meet about 3.5% of the world's electricity demand [1]. To improve the energy capture efficiency from wind of the installed wind turbines, much research has been done to find an effective MPPT control algorithm for variable-speed WECSs.

Among the existing MPPT methods, the wind speed sensorless MPPT control methods are more attractive because they do not rely on costly anemometers to acquire the wind speed information. These methods mainly include the optimal torque (OT) control, power signal feedback (PSF) control, hill-climb search (HCS) method, wind speed estimation-based control, and fuzzy logic control [2]. The OT and PSF controls are the most commonly used wind speed sensorless MPPT methods. However, either the OT curve or the optimal power curve of the wind turbine needs to be obtained through field tests first and then built into the controller. The HCS method does not need field tests and is independent of wind turbine characteristics. Therefore, it is also widely used in WECSs. Many advanced HCS methods have been proposed to solve the problems of the traditional HCS method, such as oscillation around the maximum power point (MPP) and wrong searching direction when the wind speed changes [3], [4]. However, the HCS methods have no learning ability and search for the MPPs all the times without any memory. Some research combined different MPPT techniques to enable online learning for the MPPs. For example, the reference [5] combined the HCS algorithm and the PSF control, in which the data of the maximum power versus the dc-link voltage were obtained first by using the HCS and then stored in a lookup table for online MPPT control. In that method, some memory space is required to store the lookup table, whose size depends on the power rating of the WECS. The reference [6] proposed a MPPT method with an online updating capability. That method used the HCS to search the MPPs and then used an ANN trained to memorize the MPPs. This paper aims to develop a new MPPT control method that has the online learning ability and only requires a small memory.

Q-learning is a reinforcement learning (RL) technique in the area of machine learning. Its model-free attribute enables the MPPT control algorithm to learn the MPPs without any prior knowledge of the WECS [7], [8]. However, the implementation of the traditional Q-learning requires a large lookup table to store the action values for the discretized state space, which limits its application for the problems with a large state space. To solve this dilemma, many methods have been proposed to generalize the system states using function approximation. The main methods of function approximation were discussed in [9], which include gradient-descent methods and linear methods. Reference [10] compared three different Q-learning-based algorithms,

This work was supported in part by the U.S. National Science Foundation under grant ECCS-0901218 and CAREER Award ECCS-0954938.

978-1-4799-5776-7/14/$31.00 ©2014 IEEE

including a new modified connectionist Q-learning (MCQ-L) algorithm. All of those algorithms used back-propagation ANNs. That paper concluded that the update rules of the MCQ-L method are more robust than those of the standard Q-learning.

This paper proposes a new intelligent MPPT algorithm for variable-speed WECSs based on the MCQ-L method by combining the Q-learning with ANNs. The MPPT algorithm learns the optimal speed control strategy by updating the connecting weights of the ANNs based on the received rewards, which does not require a large storage memory. Moreover, the proposed MPPT algorithm enables the WECS to behave like an intelligent agent with memory that learns to act from its own experience. The effectiveness of the proposed intelligent MPPT algorithm is verified by both PSCAD simulations for a WECS equipped with a 1.5-MW DFIG and experimental results for a WECS equipped with a 200-W PMSG.

II. ANN-BASED Q-LEARNING

Unlike the supervised learning in which an agent learns from the examples provided by an external supervisor, in the RL an agent is able to learn from its own experience by directly interacting with the environment. In a RL problem, the agent receives a numerical reward each time when it transits from one state to another. The agent's goal is to find a policy to maximize the total discounted rewards it receives over the future, which is predicted by a value function $V(\cdot)$ as follows.

$$V(s_t) = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{1}$$

where s_t is the state at time t, r_t is the reward received for the transition from state s_t to s_{t+1}, and γ□[0,1) is the discount factor that determines the current value of the rewards received in the future. Q-learning is a form of model-free RL algorithm based on temporal difference learning and has various industrial applications, partially because its convergence has been proved.

A. Q-*Learning*

The goal of Q-learning is to learn the action values (i.e., Q-values). An action value is a prediction of the total discounted reward by taking an action a from the action apace A in each state. The current action value $Q_t(s_t, a_t)$ can be updated with the predicted value function of the next state to be visited as follows,

$$Q(s_t, a_t) = r_{t+1} + \gamma V(s_{t+1}) \tag{2}$$

To maximize the overall reward, the estimation of $V(s_{t+1})$ is replaced by $\max_{a\epsilon A} Q(s_{t+1}, a)$, which is the maximum action value corresponding to the state s_{t+1},

$$Q(s_t, a_t) = r_{t+1} + \gamma \max_{a\in A} Q(s_{t+1}, a) \tag{3}$$

The one-step Q-learning is defined by

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + l_t[r_{t+1} + \gamma \max_{a\in A} Q_t(s_{t+1}, a) - Q_t(s_t, a_t)] \tag{4}$$

At each time step, the agent firstly observes its current state s_t and selects an action a_t to perform. At the same time, the Q-value $Q_t(s_t, a_t)$ is recorded. Then, the subsequent state s_{t+1} is observed with an immediate reward r_{t+1} and the maximum Q-value corresponding to s_{t+1}, $\max_{a\epsilon A} Q(s_{t+1}, a)$, is picked out. The recorded $Q_t(s_t, a_t)$ will be updated according to (4). The parameter $l_t \in (0,1]$ is the learning rate, which determines how far the currently estimated $Q_t(s_t, a_t)$ is adjusted toward the newly estimated Q-value $r_{t+1} + \gamma \max_{a\epsilon A} Q(s_{t+1}, a)$. The convergence of the Q-learning has been proved for finite-state Markova problems when a lookup table is used to store the Q-values for different state-action pairs.

B. *ANN-Based* Q-*Learning*

In the Q-learning described in Section II-A, a lookup table is required to store the values for each state-action pair, which becomes impractical for problems with a large state space. ANNs provide a powerful function approximation technique to generalize predictions between states and even provide predictions for the states never experienced before. In this paper, a multi-layer feedforward ANN with the sigmoid activation function in the hidden layer [11] is used to represent the Q-value for each action.

The Q-value for each action can be expressed as

$$Q_t = \sum_{i=1}^{n} w_{t,i}^{(2)} d_{t,i} \tag{5}$$

$$d_{t,i} = \frac{1}{1 + e^{-h_{t,i}}} \tag{6}$$

$$h_{t,i} = \sum_{j=1}^{m} w_{t,ij}^{(1)} x_{t,j} \tag{7}$$

where $w_{t,ij}^{(1)}$ and $w_{t,i}^{(2)}$ are the connecting weights between the input and hidden layers and between the hidden and output layers, respectively; $d_{t,i}$ is the output of the ith hidden node; $h_{t,i}$ is the input of the ith hidden node of the ANN; n is the total number of hidden nodes; m is the total number of inputs of the ANN.

Based on the one-step Q-learning in (4), the weight matrix w_t ($w_{t,ij}^{(1)}$ and $w_{t,i}^{(2)}$) of the ANN can be updated according to

$$\Delta w_t = \eta[r_t + \gamma \max_{a\in A} Q_{t+1} - Q_t] \nabla_w Q_t \tag{8}$$

where η is the learning rate and $\nabla_w Q_t$ is a vector of the gradient of Q_t with respect to w_t, $\partial Q_t / \partial w_t$, calculated as follows by using the back-propagation method [11].

$$\frac{\partial Q_t}{\partial w_{t,i}^{(2)}} = d_{t,i} \tag{9}$$

$$\frac{\partial Q_t}{\partial w_{t,ij}^{(1)}} = w_{t,i}^{(2)} d_{t,i}(1 - d_{t,i}) x_{t,j} \tag{10}$$

To make the training process faster, the Q-learning is combined with the temporal difference learning, and the updating rule of the weights becomes

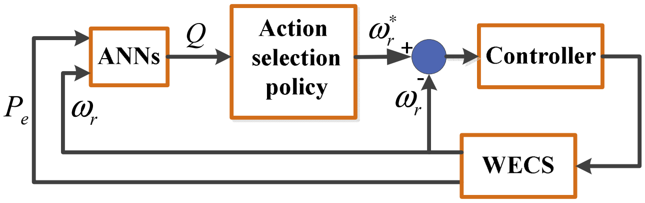


Fig. 1. Block diagram of the proposed Q-learning-based MPPT control for a WECS.

$$\Delta w_t = \eta[r_t + \gamma \max_{a\in A} Q_{t+1} - Q_t]\sum_{k=0}^{t}(\lambda\gamma)^{t-k}\nabla_w Q_k \tag{11}$$

where λ is the trace decay parameter. A larger λ means that more credit in the current reward will be given to the future states and actions. Based on (11), the MCQ-L method is proposed and the weight update rule is

$$\Delta w_t = \eta[r_t + \gamma Q_{t+1} - Q_t]\sum_{k=0}^{t}(\lambda\gamma)^{t-k}\nabla_w Q_k \tag{12}$$

Rather than using the greedy value $\max_{a\epsilon A} Q(s_{t+1}, a)$, the Q_{t+1} associated with the selected action is used for updating the weights. During the online learning process, each weight of the ANN holds an eligibility trace e_t and the weights are updated according to

$$w_{t+1} = w_t + \eta[r_t + \gamma Q_{t+1} - Q_t]e_t \tag{13}$$

For the ANN that outputs the Q-value for the selected action a_t,

$$e_t = \nabla_w Q_t + \lambda\gamma e_{t-1} \tag{14}$$

and for all other ANNs,

$$e_t = \lambda\gamma e_{t-1} \tag{15}$$

III. PROPOSED INTELLIGENT MPPT ALGORITHM

The block diagram of the proposed Q-learning-based MPPT control algorithm is shown in Fig. 1. To apply the Q-learning algorithm to the MPPT control, three items, i.e., state space, action space, and reward, should be defined properly. The agent (i.e., the WECS) observes the state ($s \in S$) of the system, i.e., the operating point of the WECS, and chooses a discrete control action ($a \in A$) from the action space. The WECS then enters into a new operating point and receives a reward r_{t+1} to update the value of the action that has been taken in the previous state. The goal of the agent is to extract as much energy as possible from the wind. To achieve this, an action with a higher value will be chosen with a greater possibility each time when the agent makes a decision to choose an action.

A. State Space

The total power that a wind turbine is able to capture from the wind can be calculated by the following formula

$$P_m = \frac{1}{2}\rho A v_w^3 C_p(\lambda, \beta) \tag{16}$$

where ρ is the air density, $A = \pi R^2$ is the area swept by the blades and R is the blade radius, v_w is the wind speed, and C_p is the power coefficient, which is determined by the tip speed ratio λ=$\omega_t R/v_w$ and the blade pitch angle β, where ω_t is the turbine rotating speed. Normally, there is an optimal value λ_{opt} at which the turbine will extract the maximum power from wind. The maximum power can be expressed by

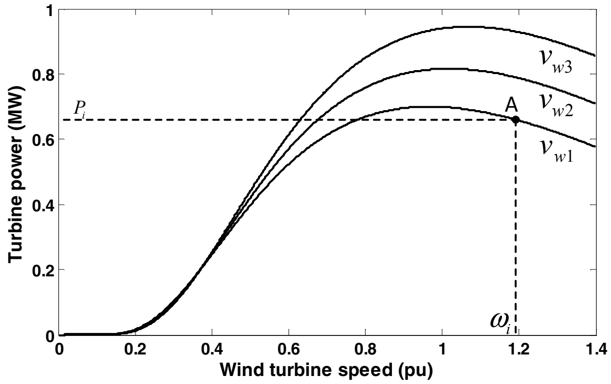


Fig. 2. Typical wind turbine power-shaft speed characteristic curves for different wind speeds.

$$P_{max} = \frac{1}{2}\rho A \frac{R^3 C_{pmax}}{\lambda_{opt}^3}\omega_t^3 = K\omega_t^3 \tag{17}$$

where C_{pmax} is the maximum power coefficient at λ_{opt} and K is the optimal parameter.

For each wind speed, there is only one optimum rotor speed at which the maximum wind power is extracted. In addition, only one rotor speed-power curve exists for a specific wind speed. As a result, a rotor speed-power pair (e.g., the point A in Fig. 2) is sufficient to represent a state, i.e., the operating point of the WECS, based on which an action could be chosen. For example, if the WECS detects that the operating point locates at A (ω_i, P_j) at wind speed $v_{\omega 1}$ by measuring its electrical output power and rotor speed, it will choose an action to move its operating point to the left to gain more energy instead of moving to the right or remaining at the point A. In real applications, the optimal relation of the electrical output power P_e and generator rotor speed ω_r is used, from which the state space is generated as follows:

$$S = \{s \mid s = (\omega_r, P_e)\} \tag{18}$$

Therefore, the measured rotor speed ω_r and electrical power P_e are the two inputs into the ANN to obtain the action values, and the total number of inputs, m, in (7) is 2 in this paper.

B. Action Space

The action space depends on the control mode of the WECS. The action space for a WECS that operates in the speed control mode can be expressed by

$$A = \{a \mid +\Delta\omega_r, 0, -\Delta\omega_r\} \tag{19}$$

where $\Delta\omega_r$ is the change of the speed control command; and 0 means that no modification is made and the previous speed control command is used. The speed control command can thus be defined as

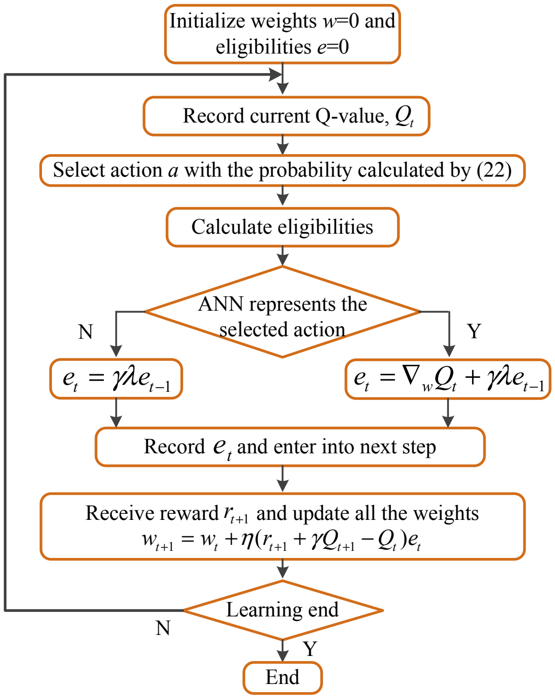

Initialize weights w=0 and eligibilities e=0
Record current Q-value, Q_t
Select action a with the probability calculated by (22)
Calculate eligibilities
ANN represents the selected action
N
Y
$e_t = \gamma\lambda e_{t-1}$
$e_t = \nabla_w Q_t + \gamma\lambda e_{t-1}$
Record e_t and enter into next step
Receive reward r_{t+1} and update all the weights
$w_{t+1} = w_t + \eta(r_{t+1} + \gamma Q_{t+1} - Q_t)e_t$
Learning end
N
Y
End

Fig. 3. Flow chart of proposed Q-learning MPPT control.

$$\omega_{r,t+1} = \omega_{r,t} + a \tag{20}$$

where $\omega_{r,t}$ and $\omega_{r,t+1}$ are the previous and new speed control commands, respectively.

C. *Reward*

After taking an action, the WECS receives a reward to evaluate the selected action. The reward function is defined as follows:

$$r_{t+1} = \begin{cases} 1, & \text{if } P_{e,t+1} - P_{e,t} > 0 \\ 0.5, & \text{if } P_{e,t+1} - P_{e,t} = 0 \\ 0, & \text{if } P_{e,t+1} - P_{e,t} < 0 \end{cases} \tag{21}$$

where $P_{e,t+1}$ and $P_{e,t}$ are two successive values of the electrical output power of the WECS. A positive reward 1 will be given to the control algorithm for selecting an action that leads to an increment of the electrical output power P_e, whereas no reward will be given when P_e decreases. If there is no change in the output power, the reward will be 0.5.

D. *Proposed MPPT Control Algorithm*

The flow chart of the proposed Q-learning MPPT control is shown in Fig. 3. The action selection policy used in this paper is Boltzmann exploration [9]. It chooses an action a in a state s with a probability according the Q-values as follows

$$p(s,a_i) = \frac{e^{Q(s,a_i)/\tau}}{\sum_{a_i} e^{Q(s,a_i)/\tau}} \tag{22}$$

where τ is a positive parameter called the temperature, which controls the randomness of the exploration. A higher temperature causes the action selection to be more random, whereas a low temperature causes the high-value actions to be selected with a greater chance. The Boltzmann exploration is implemented as follows:

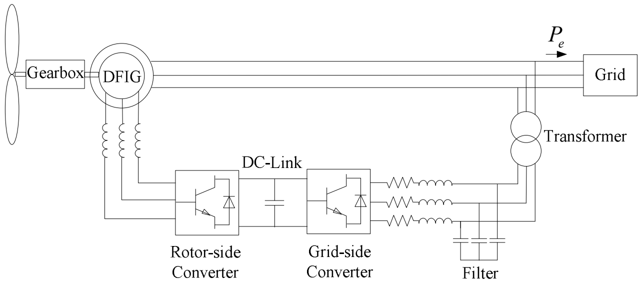


Fig. 4. Schematic of a grid-connected DFIG-based WECS.

(1) Calculate the possibility $p(s,a_i)$ for selecting each action a_i (i = 1, 2, 3) by (22).

(2) Randomly generate a number N between 0 and 1.

(3) Select an action based on N: if N is larger than $p(s,a_1)$, the first action a_1 is selected; if N is between $p(s,a_1)$ and $p(s,a_1) + p(s,a_2)$, the second action a_2 is selected; the last action a_3 will be selected if N is larger than $p(s,a_1) + p(s,a_2)$.

After initialization, the MPPT algorithm starts to learn. In each cycle of learning, $\omega_{r,t}$ and $P_{e,t}$ are measured and used as the two inputs of each ANN to calculate the current Q-values. Then an action a_t is selected from the action space defined by (19) using the Boltzmann exploration method. The eligibility of each ANN is calculated according to (14) or (15), which depends on whether the ANN represents the action that has been selected. In the next sampling period, a new state s_{t+1} will be observed and the difference of the successive electrical output power values will be used to determine the reward r_{t+1} by (21). The weights of the three ANNs are then updated according to (13).

IV. SIMULATION RESULTS

The proposed MPPT control algorithm is validated by simulation studies in PSCAD/EMTDC for a 1.5-MW DFIG-based WECS. The schematic of the WECS is shown in Fig. 4. The parameters of the DFIG and the wind turbine are presented in Appendix. The stator flux oriented vector control method is utilized to control the rotor-side converter (RSC) for decoupled control of the speed and reactive power of the DFIG. The control objective of the grid side converter (GSC) is to maintain a constant dc-link voltage and a unity power factor during the normal operation, and to provide reactive power during a grid fault if possible. The rotor speed of the DFIG can be controlled around the synchronous speed in a range of about ±30%, which enables the WECS to capture more wind energy when the wind speed changes. The reactive power commands for the RSC and GSC are set to be zero during the simulation.

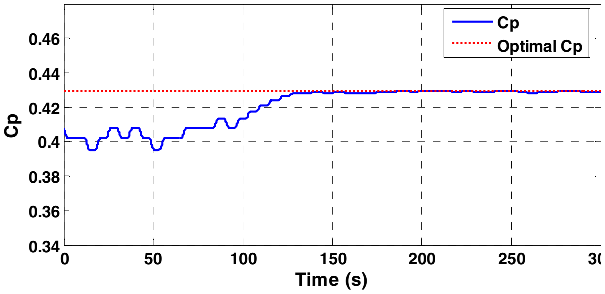


Fig. 5. Simulation result of *Cp* under a constant wind speed.

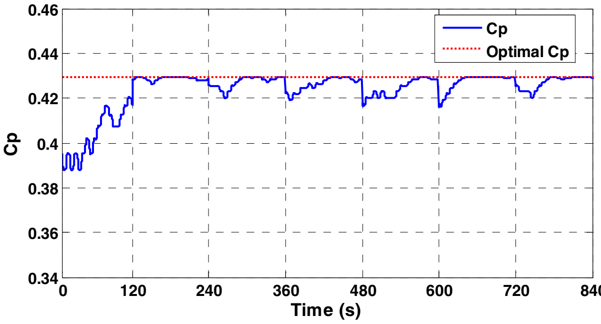


Fig. 6. Simulation results of *Cp* under a step-change wind speed.

A. Constant Wind Speed

The WECS is first operated under a constant wind speed of 9 m/s to verify that the proposed learning algorithm is capable of searching for the MPP. The simulation result of C_p is shown in Fig. 5. The WECS starts to search for the maximum C_p at 0 s and finds the MPP in about 130 s. The exploration process can be seen for the first 100 s, which means that the WECS searches for the action that produces more power. After 130 s, the WECS stays at the MPP.

B. Step-Change Wind Speed

Then, a wind speed profile with step changes (the wind speed changes alternately between 8.5 m/s and 9.5 m/s every 120 s) is used to verify that the proposed MPPT algorithm is able to learn from the experience, namely, it will reach the MPP faster when the wind speed has been experienced before for a certain time. The simulation results are shown in Fig. 6. Initially, the MPPT algorithm is very "naive" since it explores. Before 600 s, long searching periods can be seen each time wind speed changes, e.g., from 360 s to about 420 s, and from 480 s to about 560 s. In addition, the C_p does not stay at the maximum value. After 600 s, when the wind speed changes, the WECS enters into the MPP faster and stay at the optimal C_p when the MPP has been found.

V. Experimental Results

Experimental studies are performed for an emulated 200-W direct-drive PMSG-based wind turbine, as shown in Fig. 7. A 200-W DC motor is employed to emulate the dynamics of the wind turbine to drive the PMSG directly. The power generated by the PMSG is fed back to the DC source via a three-phase converter. The dual-loop vector control scheme is applied to control the rotor speed of the PMSG [12]. The overall control algorithms are implemented in the dSPACE 1104 real-time control board. All of the measured quantities are recorded via the ControlDesk interfaced with the dSPACE 1104 board and a laboratory computer (PC). The parameters of the DC motor and PMSG are listed in the Appendix. The experiment results are shown in Fig. 8.

A. Constant Wind Speed

The WECS is first operated under a constant wind speed of 8 m/s to verify that the proposed learning algorithm is capable of searching for the MPP. From Fig. 8(a), the WECS starts to search for the maximum C_p at 0 s and finds the MPP in about 60 s. The exploration process can be seen again around 70 s, from which the WECS searches for the action that produces more power. After 80 s, the WECS stays at the MPP.

B. Step-Change Wind Speed

Then, a wind speed profile with step changes (the wind speed changes alternately between 7 m/s and 8 m/s every 60 s) is used to verify that the proposed MPPT algorithm is able to learn from the experience. Initially, the MPPT algorithm is very "naive" since it explores. From Fig. 8(c), the C_p reaches the optimal value at around 270 s. After that, the C_p oscillates around the optimal value in a small range. Fig. 8(d) shows the output electrical power. At the beginning, the WECS is still learning and the output power is not the maximum. After about 270 s, the WECS generates the maximum power under the corresponding wind speed, which means that the WECS has finished its online learning process. When the wind speed changes, the WECS enters into the MPP faster, which is similar to the simulation results.

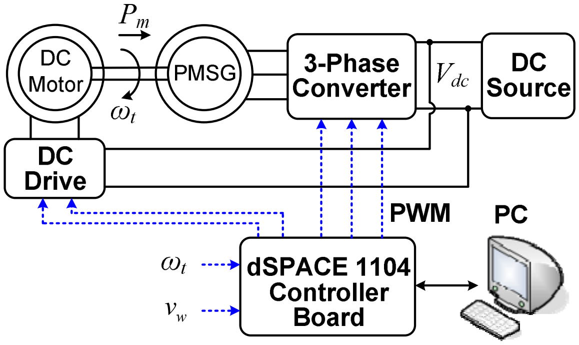
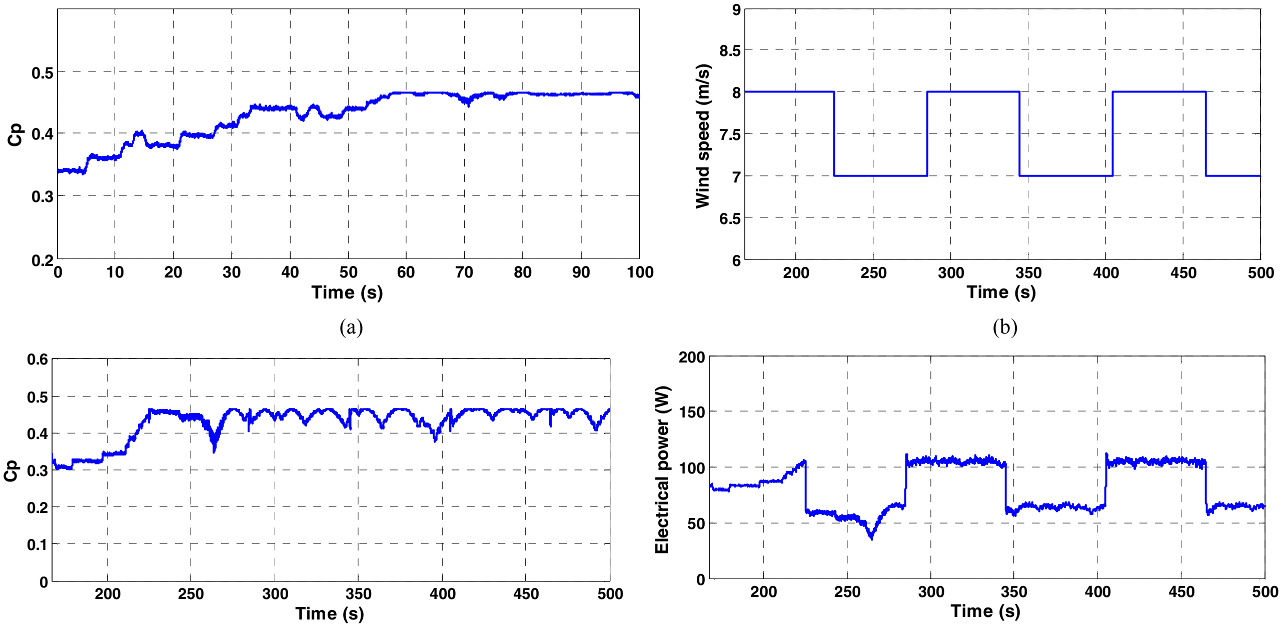


Fig. 7. The schematic of the experimental system with a WECS emulator

VI. Conclusion

This paper has proposed a new intelligent MPPT control algorithm for variable-speed WECS based on an online ANN-based Q-learning algorithm. The proposed method has been validated by PSCAD simulations for a DFIG-based



(a)

(b)

(c)

(d)

Fig. 8. Experimental results of the emulated WECS with proposed Q-learning MPPT. (a) *Cp* under a constant wind speed, (b) step-change wind speed, (c) *Cp* under the step-change wind speed, and (d) electrical output power under the step-change wind speed.

WECS and experimental results for a PMSG-based WECS emulator, where the MPPT control has been implemented in a dSPACE real-time control system. Results have shown that the proposed method enabled the WECS to learn without the knowledge of wind turbine characteristics or wind speed measurement. The computational cost has been saved by storing the information of the learned optimal control strategy in the connecting weights of ANNs. Future work can be done to improve the dynamic response and the tracking accuracy of the proposed MPPT algorithm.

Appendix

Parameters of the WECS used in simulation studies:

Wind turbine: rated power: 1.5 MW; inertia constant: 3.8 s.

DFIG: rated power: 1.5 MW; rated stator voltage: 690 V; inertia constant: 0.62 s; rated dc-bus voltage: 1.2 kV; stator resistance: 0.007 pu; wound rotor resistance: 0.009 pu; magnetizing inductance: 2.9 pu; stator leakage inductance: 0.171 pu; and wound rotor leakage inductance: 0.156 pu.

Parameters of the DC motor and PMSG used in experimental studies:

DC motor: rated speed: 3500 RPM; rated power: 200 W; back EMF constant: 8.7 V/kRPM; stator resistance: 0.39 Ω; and armature inductance 0.67 mH.

PMSG: rated speed: 3000 RPM; rated power: 200 W; number of poles: 8; back EMF constant: 9.5 V/kRPM; stator resistance: 0.233 Ω; *d*-axis inductance: 0.275 mH; and *q*-axis inductance: 0.364 mH.

References

[1] The World Wind Energy Association (WWEA). 2013 Half-year Report. Bonn, Germany. [Online]. http://www.wwindea.org/webimages/Half-year_report_2013.pdf

[2] Y. Zhao, C. Wei, Z. Zhang, and W. Qiao, "A review on position/speed sensorless control for permanent magnet synchronous machine-based wind energy conversion systems," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 1, no. 4, pp. 203-216, Dec. 2013.

[3] E. Koutroulis and K. Kalaitzakis, "Design of a maximum power tracking system for wind-energy-conversion application," *IEEE Trans. Industrial Electronics*, vol. 53, no. 2, pp. 486-494, Apr. 2006.

[4] S.M. R. Kazmi, H. Goto, H. J. Guo, and O. Ichinokura, "A novel algorithm for fast and efficient speed-sensorless maximum power point tracking in wind energy conversion systems," *IEEE Trans. Industrial Electronics*, vol. 58, no. 1, pp. 29-36, Jan. 2011.

[5] Q. Wang and L. Chang, "An intelligent maximum power extraction algorithm for inverter-based variable speed wind turbine systems," *IEEE Trans. Power Electronics*, vol. 19, no. 5, pp. 1242-1249, Sept. 2004.

[6] W. Qiao, "Intelligent mechanical sensorless MPPT control for wind energy systems," in *Proc. IEEE Power and Energy Society General Meeting*, Jul. 2012, pp. 1-8.

[7] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, pp. 279-292, 1992.

[8] C. Szepesvári, "Algorithms for reinforcement learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 4, no. 1, pp. 1-103, 2010.

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

[10] G. A. Rummery and M. Niranjan, "On-line Q-learning using connectionist systems". Eng. Dep., Cambridge Univ., Cambridge, Tech. Rep. TR-166, Sep. 1994.

[11] D. Svozil, V. Kvasnicka, and J. Pospichal, "Introduction to multi-layer feed-forward neural networks," *Chemometrics and Intelligent Laboratory Systems*, vol. 39, no. 1, pp. 43-62, Nov. 1997.

[12] W. Qiao, X. Yang, and X. Gong, "Wind speed and rotor position sensorless control for direct-drive PMG wind turbines," *IEEE Trans. Ind. Appl.*, vol. 48, no. 1, pp. 3-11, Jan/Feb 2012.