Jaen Sola, Pablo and Anaya-Lara, Olimpo and Dominguez-Navarro, Jose Antonio (2013) Adaptive dynamic control of a wind generator. In: 9th PhD Seminar on Wind Energy in Europe, 2013-09-18 - 2013-09-20, Uppsala University Campus Gotland. (In Press) ,

Adaptive Dynamic Control of a Wind Generator

Pablo Jaen Sola*, Olimpo Anaya-Lara, Jose Antonio Dominguez-Navarro
CDT for Wind Energy Systems, University of Strathclyde, Glasgow, UK
*e-mail: pablo.jaen-sola@strath.ac.uk

ABSTRACT

The main aim of the present project is to obtain a better understanding of the dynamic control of a wind turbine. To accomplish this, an adaptive dynamic controller was developed and connected to a wind DFIG Simulink model. In order to demonstrate the effectiveness of the controller in question, a three phase fault in the power grid was simulated. Results showed how the implementation of the adaptive dynamic programming controller helped to improve the speed of response of the system. It also helped to reduce the magnitude of the oscillations taking place after the disturbance is removed.

The non-linear systems are difficult to model and to control. New modern control techniques allow controlling these systems without having the system model, reducing the complexity. Among these techniques, it is the adaptive dynamic control that estimates the parameters of the dynamic system on-line and adapts their control to the new conditions without spreading the error throughout a feedback network.

INTRODUCTION

The fossil fuel resources on the earth are limited and the oil, gas and coal production will peak in a few decades. This has woken up the interest in green energies, with the wind energy industry playing an important role. Since wind is an intermittent and random resource control strategies are needed to maximize energy capture. This paper focuses on adaptive dynamic programming.

Three different types of wind turbine generator technologies exist: fixed-speed induction generator (FSIG), doubly fed induction generator (DFIG) and fully-rated converter wind turbine (FRC). In order to develop decoupled control of the active and reactive power, the DFIG model is required. The DFIG is used worldwide due to its advantageous characteristics. More control flexibility, improvement of power quality and system efficiency and exploitation of features provided by wind turbine power electronics stand out. The stator of the DFIG is directly connected to the grid while to connect the rotor a back to back converter is needed. The said converter needs to be rated for a fraction of the total output power. This fraction depends on the allowable sub- and super-synchronous speed range. The converter consists of three main components: rotor side controller, grid side controller and DC link capacitor. The use of the said controller ensures constant voltage and constant frequency to the grid as well as good power quality. Previous studies demonstrated that the controller of the back to back converter significantly affects the stability of the DFIG [1].

The implementation of a STATCOM was studied in [2]to help with the uninterrupted operation of a wind farm having DFIGs during power grid disturbances. The power network utilized in [2] is a single machine infinite bus system and there is no coordination between the wind farm and the STATCOM. In [3], the authors used a heuristic dynamic programming technique to design a novel interface neurocontroller for the coordinated reactive power control between a large wind farm composed by DFIGs and a STATCOM obtaining fair results.

In [4], the authors proposed an on-line learning control system based on neural dynamic programming. This on-line control improves the performance of the plant over time due to it learns from its own errors and tries to strength its signal to enhance its future performance. It also memorizes system states with positive reinforcement. In future, similar states will be associated with a control action leading to a positive reinforcement.

In this paper, the adaptive dynamic programming (ADP) architecture proposed in [4] and utilized in [5] has been used to control the reactive power of a wind farm under a three phase grid fault. Two neural networks, the action and the critic, were programmed in a separated Matlab function and connected to the power_wind_dfig phasor Simulink model. Simulation studies were carried out to verify the controller. As mentioned, the main aim of this project is to obtain a better understanding of the dynamic control of a wind generator. To do so a replica of the application developed in [5] has been done.

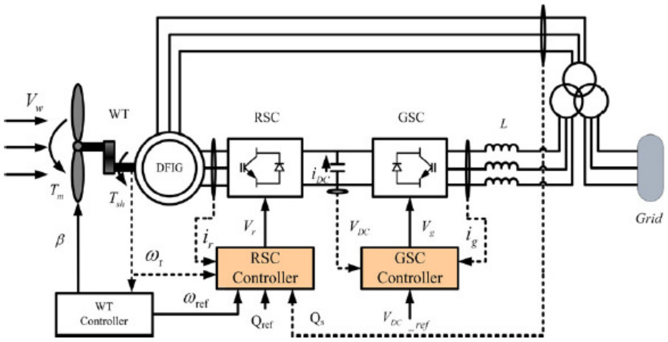

Figure 1. Wind turbine power system [5]

WIND FARM POWER SYSTEM

Fig. 1 illustrates the wind turbine power system used in this paper. The wind turbine is connected to the DFIG through a low speed shaft, a gearbox and a high speed shaft respectively.

The wind farm has been represented by this one large wind turbine with a DFIG generator. 30MW are produced by this wind farm consisting of twenty 1.5MW rated power wind turbines. Fig. 2 shows the model used in the simulation studies with the controller connected.

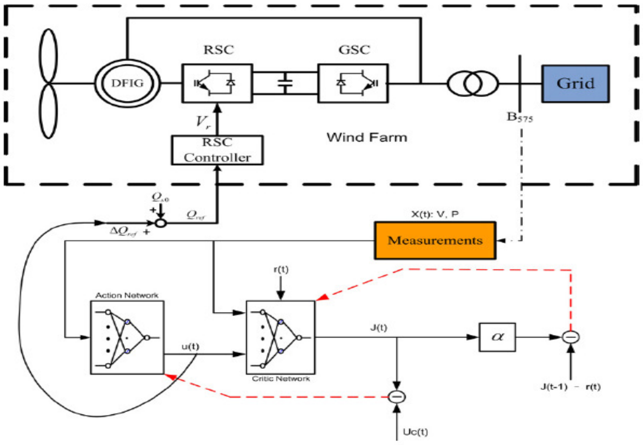

Figure 2. Wind farm model [5]

WIND FARM MODEL WITH ADP CONTROLLER

Fig. 3 displays a sketch of the layout of the on-line ADP controller [4]. The reinforcement signal, r(t), is externally provided and depends on voltage and active power measurements from the wind farm. X is the 2 elements state input vector and u is the control signal. Note that signal u is also one of the 5 inputs of the critic network. The output of the critic network, J, approximates the discounted reward-to-go value of the Bellman equation [6]. Specifically, it approximates R(t)at time t as follows,

$$R(t) = \sum_{k=1}^{\infty} \alpha^{k-1} r(t+k) \tag{1}$$

Where R(t) is the future accumulative reward-to-go value at time and α is a discount factor for the infinite-horizon problem (0 < α < 1). In this paper, α is equal to 0.95.

At the beginning, the ADP controller is "naive". Weights and parameters in both networks are initialized using random values. After the first observation of the system, parameters start adjusting according to the situation in the action network.

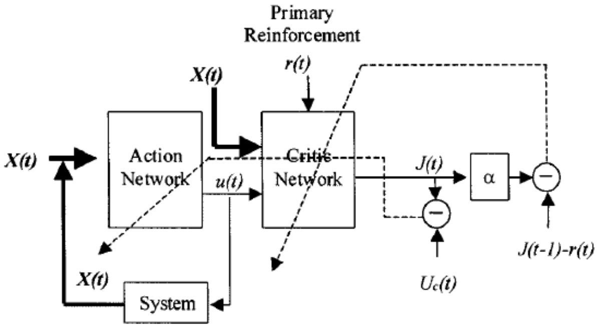

Figure 3. Layout of the ADP controller [4]

A. The Critic Network

The critic network adapts to approximate the output function J(t). Equation 2 is used to predict the error in the critic network.

$$e_c(t) = \alpha J(t) - [J(t-1) - r(t)] \tag{2}$$

In this network the objective function to be minimized is,

$$E_c(t) = \tfrac{1}{2} e_c^2(t) \tag{3}$$

The weights are updated by making use of a gradient-based adaptation given by the next equations,

$$w_c(t+1) = w_c(t) + \Delta w_c(t) \tag{4}$$

$$\Delta w_c(t) = l(t)\left[-\frac{\partial E_c(t)}{\partial w_c(t)}\right] \tag{5}$$

$$\frac{\partial E_c(t)}{\partial w_c(t)} = \left[\frac{\partial E_c(t)}{\partial J(t)} \frac{\partial J(t)}{\partial w_c(t)}\right] \tag{6}$$

Where l(t) > 0 is the learning rate at time t. This parameter has been maintained constant during the whole process with a value of 0.3. w_c is the matrix of weights in the critic network.

B. The Action Network

The idea behind the ADP controller is to backpropagate the error between the target, U_c, and the approximation of the J function [7]. In this paper, U_c has been set to 0. To update the network weights the following equations have been utilized. Equation 7 corresponds to the error in the action network while Equation 8 shows the performance error measure. The point in updating the weights is to minimize this performance error.

$$e_a(t) = U_c(t) - J(t) \tag{7}$$

$$E_a(t) = \tfrac{1}{2} e_a^2(t) \tag{8}$$

The algorithm to update the weights in the action network is very similar to that of the critic network. Its gradient descent policy is,

$$w_a(t+1) = w_a(t) + \Delta w_a(t) \tag{9}$$

$$\Delta w_a(t) = l(t)\left[-\frac{\partial E_a(t)}{\partial w_a(t)}\right] \tag{10}$$

$$\frac{\partial E_a(t)}{\partial w_a(t)} = \frac{\partial E_a(t)}{\partial J(t)} \frac{\partial J(t)}{\partial u(t)} \frac{\partial u(t)}{\partial w_a(t)} \tag{11}$$

where w_a is the matrix of weights in the action network and l(t) is the learning rate.

C. Learning Algorithms

In the learning system described above, two elements stand out, the critic network and the action network. These non-linear multilayer feedforward networks have one hidden layer with 6 neurons. The approximation for the J function in the critic network is,

$$q(t) = w_c^{(1)}(t) x(t) \tag{12}$$

$$p(t) = \frac{1 - \exp^{-q(t)}}{1 + \exp^{-q(t)}} \tag{13}$$

$$J(t) = w_c^{(2)}(t) p(t) \tag{14}$$

Where

q inputs of the hidden layer in the critic network

p outputs of the hidden layer in the critic network

x state vector

Since all the outputs of the hidden layer converge in the node of the last layer and we have worked with matrices, all the elements within the obtained 1x6 matrix have to be summed together to obtain a scalar value of J.

The adaptation of the critic network is,

- $\Delta w_c^{(2)}$ (hidden layer to output layer)

$$\Delta w_c^{(2)}(t) = l(t)\left[-\frac{\partial E_c(t)}{\partial w_c^{(2)}(t)}\right] \tag{15}$$

$$\frac{\partial E_c(t)}{\partial w_c^{(2)}(t)} = \alpha e_c(t) p(t) \tag{16}$$

- $\Delta w_c^{(1)}$ (input layer to hidden layer)

$$\Delta w_c^{(1)}(t) = l(t)\left[-\frac{\partial E_c(t)}{\partial w_c^{(1)}(t)}\right] \tag{17}$$

$$\frac{\partial E_c(t)}{\partial w_c^{(1)}(t)} = \alpha e_c(t) w_c^{(2)}(t) \left[\tfrac{1}{2}\left(1 - p^2(t)\right)\right] x(t) \tag{18}$$

The adaptation in the action network is similar to that of the critic network. It is also implemented by a feedforward network with a 6 neurons hidden layer. However, this network only has 4 inputs. As said, the equations for the action network are similar to those of the critic network with u being the input to the action node. Since in this paper, we have worked with matrices, all the elements in the matrix 1x6 that arrive into the action node have to be summed together so as to obtain a scalar value of the output of the action network. h(t) is the input vector to the hidden layer while g(t) is the output vector of the said layer. The rule to update the weights in the action network also owns two sets of equations.

- $\Delta w_a^{(1)}$ (hidden layer to output layer)

$$\Delta w_a^{(2)}(t) = l(t)\left[-\frac{\partial E_a(t)}{\partial w_a^{(2)}(t)}\right] \tag{19}$$

$$\frac{\partial E_a(t)}{\partial w_a^{(2)}(t)} = e_a(t)\left[\tfrac{1}{2}\left(1 - u^2(t)\right)\right] g(t) \left[w_c^{(2)}(t)\, \tfrac{1}{2}\, (1 - p^2(t)) w_c^{(1)}(t)\right] \tag{20}$$

The last term of Equation 22 corresponds to the matrix of weights associated with the input element from the action network.

- $\Delta w_a^{(2)}$ (input layer to hidden layer)

$$\Delta w_a^{(1)}(t) = l(t)\left[-\frac{\partial E_a(t)}{\partial w_a^{(1)}(t)}\right] \tag{21}$$

$$\frac{\partial E_a(t)}{\partial w_a^{(1)}(t)} = e_a(t)\left[\tfrac{1}{2}\left(1-u^2(t)\right)\right] w_a^{(2)}(t) \tag{22}$$
$$\left[\tfrac{1}{2}\left(1-g^2(t)\right)\right] x(t)\left[w_c^{(2)}(t)\,\tfrac{1}{2}\,(1-p^2(t)) w_c^{(1)}(t)\right]$$

As the weights are initialized randomly, they need to be normalized in order to restrict them into a desired range. Norm 1 was used.

$$w_c(t+1) = \frac{w_c(t) + \Delta w_c(t)}{\|w_c(t) + \Delta w_c(t)\|_1} \tag{24}$$

The equation to normalize the weights in the action
is similar to the equation used to the critic network.

In the critic network, Equations 16 and 19 were utilized to update the weights while Equations 20 and 22 were used for updating the weights in the action network.

D. Design of the ADP based wind farm controller

A 3 elements state vector contains the voltage, V, the active power, P, both measured at bus 575 and the desired ultimate objective, U_c, that in our case is 0. This vector is the input to our ADP controller. The output of the controller, u, is an additional signal ΔQ_{ref}, which is added to the steady state signal Q_{s0} to produce the control signal Q_{ref}. The said control signal is then connected to the rotor side controller of the converter. The main aim of controlling the reference reactive power signal is that the generator can produce or absorb reactive power when needed to keep the DFIG voltage level at its corresponding level. If a large disturbance in the power grid takes place, the Q_{ref} signal will be able to adjust allowing the DFIG either to absorb or to produce reactive power depending on the features of the fault. This adjustment can significantly improve the behavior of the system after the grid fault has been removed by reducing the sag and the overshot of the voltage of the wind farm and at the cross coupling point. It also helps damp the oscillations after the fault is recovered.

Under steady state conditions, Q_{ref} is considered to be reactive neutral by only introducing a constant signal of value 0.

E. Design of the critic network

As described, it is a multilayer network with 6 neurons in the hidden layer. To define the output of the hidden nodes a sigmoid function has been selected. A 5 elements vector containing the measured states of the voltage, V, and active power, P, at bus 575, their one time-delayed values and the output of the action network, u or Q_{ref}, is input to the critic network. Signal r(t) is not an input of the network. However, it is used to define the error function. The reinforcement signal is as follows,

$$\begin{aligned} r(t) = \tfrac{1}{2}\left[(V(t)-1.026)^2 + 0.6(V(t-1)-1.026)^2\right] \\ +\tfrac{1}{2}\left[(P(t)-0.53)^2 \right. \\ \left. + 0.6(P(t-1)-0.53)^2\right] \end{aligned} \tag{25}$$

Where 1.026 and 0.53 are an approximation of the nominal values per unit of voltage and active power respectively.

The output of the critic network is an approximation of the value of the Bellman equation.

F. Design of the action network

As defined, the architecture of both networks is really similar. With the same number of hidden nodes defined by sigmoid functions, this multilayer network has a 4 elements input vector. Voltage, V, and active power, P, at bus 575 and their one time-delayed values form the said vector. The output of the network is the control signal ΔQ_{ref}. A linear function was used to define the output of the last layer.

SIMULATION RESULTS

In [8], the authors demonstrated that there is no interaction between the wind turbines of a wind farm if their controllers are properly tuned. Having this in mind, our wind farm has been represented by a large wind turbine with a DFIG generator. As it has been already shown in Fig. 2, our infinite bus system produces 30MW through twenty 1.5MW wind turbines. The wind turbines are connected to a 25kV distribution system that exports power to a 120kV grid through a 30km, 25kV feeder. To the wind farm, the 120kV grid is equal to an infinite bus. All the turbines in the farm have DFIG technology consisting of wound rotor induction generators and back to back converters. The stator is directly connected to the 60Hz power grid while the rotor is fed at a variable frequency through the mentioned converter [9].

Following the guidelines in [4], the ADP was developed and connected to the converter to verify its effectiveness. A 75ms three phase fault was simulated. The wind speed was varied from 8 to 14m/s with a step time of 5 seconds. The duration of the simulation was set to 30 seconds with the disturbance taking place at 25 seconds. The wind turbine protection was disabled. A comparison of the performance of the wind farm with and without the ADP controller was carried out. Figs. 4, 5, 6 and 7 display the voltage of the wind farm at bus 575 with and without controller and the voltage at the cross coupling point (25kV) with and without ADP controller respectively. As seen, the behavior of the system after the disturbance has been considerably improved. The speed of response of the voltage signal at bus 575 and at the cross coupling point has been increased. The oscillations after the grid is recovered have been damped as well.

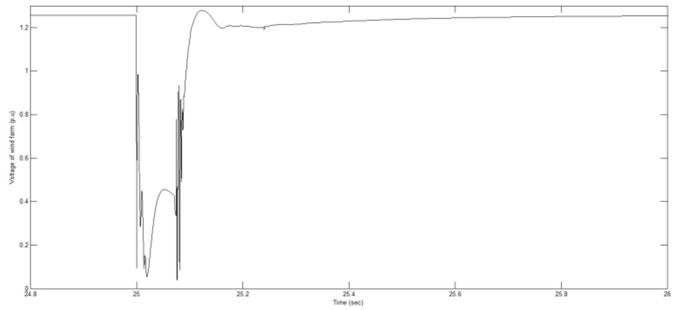

Figure 4. Wind farm voltage with ADP at bus 575

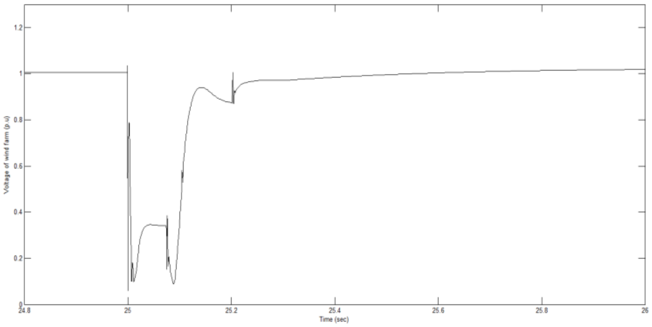

Figure 5. Wind farm voltage without ADP at bus 575

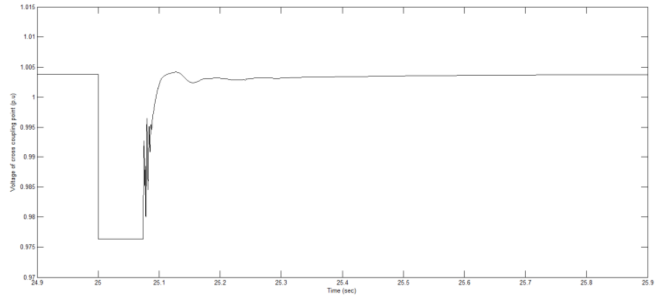

Figure 6. Voltage with ADP at the cross coupling point

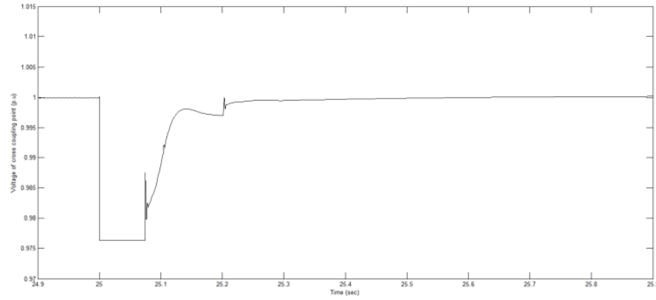

Figure 7. Voltage without ADP at the cross coupling point

DISCUSSION AND CONCLUSION

This paper tackles DFIG generator reactive power control from the ADP perspective. The architecture of the model used in our simulations studies has been shown in detail. A detailed description of the controller utilized and how it was developed has been provided too. According to the results acquired, it can be said that the performance of the wind turbine under large disturbances can be improved by making use of the ADP controller. The speed of response of the voltage signals was considerably enhanced. The damping features of the signals were also significantly improved. However, certain differences between the results obtained in this study, with and without the ADP controller, and the results given by the authors in [5] have been observed even though the same approach has been followed.

In future, more complex and larger models could be studied in order to verify the effectiveness of our controller on these scenarios. It would be also interesting to vary parameters such as wind speed or learning rate to see how the controller behaves.

REFERENCES

[1] R. Pena, J. Clear and G. Asher, "Doubly fed induction generator using back to back PWM converter and its application to variable-speed wind energy generation," IEEE Proc.: Power Appl., vol. 143, no. 3, pp. 231-241, 1996.

[2] W. Qiao, G. Venayagamoorthy and R. Harley, "Real time implementation of a STATCOM on a wind farm equipped with doubly fed induction generators," in Proc. IEEE IAS 41st Annu.Meet., Tampa, FL, 2006.

[3] W. Qiao, R. Harley and G. Venayagamoorthy, "Coordinated control reactive power control of a large wind farm and a STATCOM using heuristic dynamic programming," IEEE Trans. Energy Convers., vol. 24, no. 2, pp. 98-107, 2009.

[4] J. Si and Y. Wang, "Online learning control by association and reinforcement," IEEE Trans. Neural Networks, vol. 12, no. 2, pp. 264-276, 2001.

[5] Y. Tang, H. He, Z. Ni and X. Sui, "Reactive power control of grid-connected wind farm based on adaptive dynamic programming," Neurocomputing, 2013.

[6] R. Bellman and S. Dreyfus, Applied Dynamic Programming, Princeton, NJ: Princeton Univ. Press, 1962.

[7] J. Si and D. Liu, "Direct neural dynamic programming, in: Handbook of Learning," IEEE Press, p. 125–151, 2004.

[8] A. V., "Analysis of Dynamic behavior of Electric Power Systems with Large Amount of Wind Power, PhD Dissertation," Technical University of Denmark, Kgs. Lyngby, Denmark, April 2003.

[9] Hydro-Quebec, and The MathWorks, Inc., "Wind Farm (DFIG Phasor Model)," Matlab R2012b, Copyright 1997-2012.