IET Journals
The Institution of Engineering and Technology

Model-Free Adaptive Learning Control Scheme for Wind Turbines with Doubly Fed Induction Generators

ISSN 1751-8644
doi: 0000000000
www.ietdl.org

Mohammed Abouheaf[1*], *Wail Gueaieb*[1], *Adel Sharaf*[2]

[1] *School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Ontario, Canada*
[2] *SHARAF Energy Systems, Inc. , Fredericton, New Brunswick, Canada*
[*] *E-mail: mohammed.abouheaf@uottawa.ca*

Abstract: The classical control mechanisms of the wind turbines are generally based on precise modeling approaches to ensure robust and effective interplay between the wind turbines and the main power grids in both autonomous and grid-connected modes. The paper presents an innovative intelligent control system for the doubly fed induction generator wind turbines. The proposed system uses model-free control polices. The online controller is based on a policy iteration reinforcement learning paradigm along with an adaptive actor-critic technique. It is shown to be robust against the turbine's high nonlinearities and stochastic variations in the input-output conditions. These are associated with single and double rotor doubly fed large scale induction generators driven by wind turbines in the range of 5-7 MW. The performance of the controller is validated against challenging scenarios of coexisting undesired situations like severe wind changes with load excursions and abrupt shifts in the loads.

Nomenclature

J	Inertia constant of the wind turbine.
s	Rotor slip.
f_b, ω_b	Base frequency and base angular frequency.
ω_r, ω_s	Rotor and stator angular frequencies.
T_E, T_M	Electromagnetic torque and mechanical torque.
d, q	Subscripts indicate the direct and quadratic components.
r, s	Subscripts indicate the rotor and stator values.
R_r, R_s	Self rotor and stator resistances.
L_{rr}, L_{ss}	Rotor and stator self inductance.
L_m	Magnetizing inductance.
L_{rm}	Mutual inductance between two rotor coils.
L_{dd}	Double-Cage self inductance.
$i_{dr}, i_{ds}, i_{dd}, i_{qr}, i_{qs}, i_{qd}$	Rotor, stator, and double-cage currents in the dq frame.
v_r, v_{sr}	Rotor and stator internal voltages.

1 Introduction

The limited fossil fuel resources and their impact on the environment urged the research societies and the industrial entities to develop effective and reliable technologies to integrate the renewable generation sources into the main power grids. The power system networks are highly nonlinear systems with continuous varying dynamics that involve large number of electrical and mechanical systems. Although the renewable energy technologies have achieved great advances during the last three decades, their integration into the main power grids creates remarkable degradation in the operation of the power system networks. This is due to the lack of reliable and accurate modeling approaches to cope with the continuous varying dynamics of the large-scale power system networks. Thus, the need to develop reliable model-free control schemes is highly appreciated. This paper introduces an online model-free adaptive learning controller for the variable speed wind turbines. This approach does not use any of the dynamics of the wind turbines. The proposed adaptive learning controller is shown to be robust against the uncertainties in the dynamics and the severe wind changes with load excursions.

Many large wind farms employ Doubly Fed Induction Generator (DFIG) variable speed wind turbines because of their compatibility with the power system networks and their abilities to reduce the mechanical loads [1–3]. The main interest in DFIG-based wind turbines is due to their efficiency, power quality, and controllability [4, 5]. Accurate DFIG dynamical models are necessary in order to develop the control algorithms and protection systems for the variable wind turbines [2]. The modeling of the variable speed wind turbine with multiple pole synchronous generators in dynamic simulation environment is introduced in [6]. The dynamical modeling of a class of variable speed wind turbines based on synchronous generators is introduced in [7]. These models involve non-reduced order models that consider both rotor and stator transients and reduced order models that overlook the stator transients. Reduced order models for single-cage based DFIG wind turbines are developed in [8, 9]. Fundamental frequency-based wind turbine models are introduced in [4]. Dynamic models for the DFIG wind turbines based on single induction generators (SCIG) and double-cage induction generators (DCIG) are introduced in [2]. Different state space models for the variable speed wind turbines are highlighted in [10]. A model that represents variable speed wind turbine in power system simulations is considered in [11]. A survey about the different types of the wind generator systems is introduced in [12].

The stability of large wind farms is studied using a control mechanism that uses voltage control network for the stator and rotor converters [2]. The frequency regulation ability for the DFIG wind turbines is assessed in [13], where the inertia support by the DFIG wind turbine was shown to be possible by adding a control loop in the inertia contributing loop. A fractional order control strategy is proposed for the variable-speed wind turbines with permanent magnet synchronous generators and full-power converter topology [14]. An energy management system based on a fuzzy fractional order PID controller is developed in [15]. This control scheme is able to analyze and simulate the dynamic behavior in the grid-connected and isolated modes. A two-layer control scheme is developed for DFIG wind turbine in [16]. The lower layer, controls the wind turbines

with energy storage systems. The second layer, works a supervisory control structure for the wind farm [16]. A current control approach based on positive/negative reference frames is developed to control the rotor sequence currents in [17]. A control scheme is developed for DFIG wind turbine in [18], this controller is able to provide voltage support, robust fault recovery, and short-term frequency support during power generation losses. The sudden drop in the grid's voltage results in short-circuit currents that lead to an electromotive force in the rotor side of the DFIG wind turbine, which could severely damage the converter and the rotor [29, 30]. The crowbar provides a protection mechanism to control the surge currents which occur during the Low-Voltage-Ride-Through (LVRT) [28]. The transient flux characteristics of the DFIG wind turbines with crowbar are studied in [28]. The crowbar is introduced to limit the short circuit currents once the fault occurs [31].

The optimal control problem is formulated in the frame-work of Dynamic Programming [19, 20]. The Dynamic Programming problems are solved using the different approaches of Approximate Dynamic Programming (ADP) [13, 20, 21]. The ADP solution methods are based on selecting the solution structures (the value functions) and the way the optimal policies are updated [23]. The optimal control problems for the dynamical systems are formulated as decision processes in the Artificial Intelligence (AI) framework. ADP brings together Dynamic Programming, Reinforcement Learning (RL), and Adaptive Critics to solve the optimal control problems [21–23]. The RL approaches approximate the value function structures in a dynamic environment using reward or penalty objective functions, and the usefulness of the taken policies is assessed using the utility functions [21–23]. Reinforcement Learning (RL) techniques are developed online using two-step methods known as Value Iteration (VI) or Policy Iteration (PI). These techniques are implemented using the actor-critic neural network structures. The value functions are approximated using critic neural network structures, while the optimal policies are approximated using the actor neural network structures [23]. The actor approximates the optimal decision and applies it to the dynamic environment and then the quality of this decision is assessed and approximated by the critic neural network structure [23]. The optimal control solution finds the optimal policies and the optimality conditions which relate the optimal value function to the optimal strategy [24]. An online value iteration based control system is developed to control an autonomous smartgrid in [25]. This technique is implemented in real-time using partial knowledge about the smartgrid's dynamics. A novel online adaptive control scheme is developed for the distributed smartgrids working under stochastic disturbances in [26]. The control approach is implemented using means of the adaptive critics. A frequency regulation approach is proposed to coordinate the inertial, rotor speed, and pitch angle control mechanisms under different wind speed modes in [27].

The current manuscript contributes to the state of the art with an innovative online adaptive learning control system for variable speed wind turbines, where the learning scheme and the optimal control policy do not require any prior knowledge of the dynamics of the DFIG-equipped wind turbine. This control approach can be easily generalized to different models with similar characteristics, since it treats the plant (wind turbine) as a black box. The adaptive learning approach uses a real-time policy iteration mechanism, where the convergence is guaranteed if the initial policy is admissible. The adaptive learning algorithm is implemented by means of a single layer of neural network structures with smooth tuning laws. The proposed approach guarantees the closed-loop system's asymptotic stability as well as better damping characteristics compared to those achieved using the classical solution of the linear quadratic regulator (Riccati solution).

The paper is organized as follows. Section 2 presents the dynamical models for the single and double-cage induction generator wind turbines. Section 3 lays out the mathematical foundation for the underlying model-free optimal control problem. Section 4 introduces the adaptive learning control algorithm using a policy iteration process along with its convergence proof. Section 5 shows the online implementation of the adaptive learning controller using the adaptive critics. Section 6 displays the validation results of the intelligent control algorithm and compares its performance to the classical Riccati solution. Section 7 concludes the paper with a few concluding remarks and possible future research avenues.

2 DFIG Wind Turbine

The structure of the DFIG-equipped wind turbine includes wound rotor induction generator with a slip ring that couples the rotor currents to two back-to-back voltage-fed pulse width modulated converters in two directions as shown in Figure 1.

This structure is able to control the magnitude and the direction of the power fed to the AC-supply network, consequently the DFIG-equipped wind turbine works in a variable speed operation mode. The speed of the wind turbine is controlled by managing the switching signals of the insulated gate bipolar transistors [4]. On the other hand, the fixed speed wind turbines have (1-2%) rotor speed variations [4]. The crowbar circuit is used to provide over-current and voltage-limit protection. The following standard assumptions are considered in order to model the induction generators [2];

* The positive direction of the stator current is the direction toward the machine.
* The synchronous frame is considered the reference frame.
* The dynamical equations are expressed in the *dq*-reference frame.
* The stator transients are neglected.
* The higher order rotor currents are neglected.

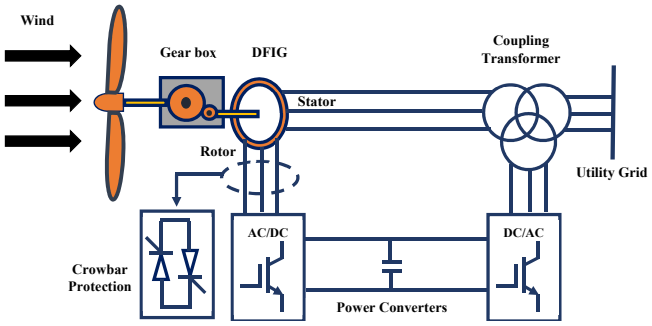

Fig. 1: Schematic diagram of DFIG-equipped wind turbine.

In the sequel, two induction generator models (single and double-cage) are considered to study the closed-loop control characteristics of the DFIG-equipped wind turbines [2, 10]. These models are used to generate the necessary measurements to validate the performance of the adaptive learning controller. All the variables are expressed in per-unit values.

2.1 Single-Cage Induction Generator (SCIG):

The following setup for the single-cage induction generator model is useful to develop the *PV* based control structures [2, 10]. The states for this model are the (stator/rotor) currents, the control inputs are the (stator/rotor) voltages, and the outputs are the rotor currents such that $\boldsymbol{X}^{SI} = [i_{ds}\ i_{qs}\ i_{dr}\ i_{qr}]^T$, $\boldsymbol{u}^{SI} = [v_{ds}\ v_{qs}\ v_{dr}\ v_{qr}]^T$, and $\boldsymbol{y}^{SI} = [i_{dr}\ i_{qr}]^T$ [2, 10].

The state space model for the SCIG case is given by

$$\begin{array}{rcl} \dot{\boldsymbol{X}}^{SI} & = & \boldsymbol{A}^{SI}\,\boldsymbol{X}^{SI} + \boldsymbol{B}^{SI}\,\boldsymbol{u}^{SI}, \\ \boldsymbol{y}^{SI} & = & \boldsymbol{C}^{SI}\,\boldsymbol{X}^{SI} + \boldsymbol{D}^{SI}\,\boldsymbol{u}^{SI}, \end{array} \tag{1}$$

where $\boldsymbol{A}^{SI}, \boldsymbol{B}^{SI}, \boldsymbol{C}^{SI}$, and $\boldsymbol{D}^{SI}$ are the values of the state space matrices for the single-cage induction generator.

In order to complete the dynamical model, the mechanical model of the rotor or the machine swing equation is given by [10]

$$\dot{\omega}_r = \frac{1}{2\,J}\left(T_M - T_E\right), \quad T_E = L_m\left(i_{dr} i_{qs} - i_{qr} i_{ds}\right).$$

2.2 Double-Cage Induction Generator (DCIG):

The following setup for double-cage induction generator model is useful when different power system disturbances exist and it is required to have an accurate model that takes into consideration the machine transient and sub-transient characteristics.

The states for this model are the (stator/rotor/double-cage) currents, the control inputs, are the (stator/rotor) voltages, and the outputs are the rotor currents such that $\boldsymbol{X}^{DI} = \left[i_{ds}\ i_{qs}\ i_{dr}\ i_{qr}\ i_{dd}\ i_{qd}\right]^T$, $\boldsymbol{u}^{DI} = [v_{ds}\ v_{qs}\ v_{dr}\ v_{qr}]^T$, and $\boldsymbol{y}^{DI} = [i_{dr}\ i_{qr}]^T$ [2, 10].

The state space model for the double induction case is given by

$$\begin{array}{rcl} \dot{\mathbf{X}}^{DI} & = & \boldsymbol{A}^{DI}\,\boldsymbol{X}^{DI} + \boldsymbol{B}^{DI}\,\boldsymbol{u}^{DI}, \\ \boldsymbol{y}^{DI} & = & \boldsymbol{C}^{DI}\,\boldsymbol{X}^{DI} + \boldsymbol{D}^{DI}\,\boldsymbol{u}^{DI}, \end{array} \tag{2}$$

where $\boldsymbol{A}^{DI}$, $\boldsymbol{B}^{DI}$, $\boldsymbol{C}^{DI}$, and $\boldsymbol{D}^{DI}$ are the values of the state space matrices for the double-cage induction generator turbine.

The mechanical model of the rotor or the machine swing equation is given by [2]

$$\dot{\omega}_r = \frac{1}{J}\left(T_M - T_E\right), \quad T_E = v_{qs} i_{qs} + v_{ds} i_{ds}.$$

3 The Optimal Control Problem Formulation

This section presents the mathematical foundation for the underlying optimal control problem. First, Bellman equation and the associated optimality conditions are found using the classical Bellman control theory [24]. Then, a modified version of Bellman equation is considered to account for the value function dependency on both the states and the control policy of the wind turbine. This structure motivates the hence-coming development for having an online solving structure with model-free optimal policies.

3.1 Bellman Equation

The optimal control problem is solved in the discrete-time frame. Thus, the dynamical equations for the different types of the DFIG-operated wind turbines are expressed in the following form

$$\boldsymbol{X}_{k+1} = \boldsymbol{A}\,\boldsymbol{X}_k + \boldsymbol{B}\,\boldsymbol{u}_k, \tag{3}$$

where $\boldsymbol{X} \in R^n$ and $\boldsymbol{u} \in R^m$ are vectors of the states and the input control signals respectively. The matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are the discretized state space matrices for each wind turbine type.

The optimal control problem is formalized as an optimization problem, where an objective function is minimized, in order to have the optimal control policies. Thus, a quadratic performance index is proposed to assess the usefulness of the taken control policies so that

$$P = \sum_{\ell=0}^{\infty} U(\boldsymbol{X}_\ell, \boldsymbol{u}_\ell), \tag{4}$$

where U is a quadratic convex cost function given by

$$U(\boldsymbol{X}_\ell, \boldsymbol{u}_\ell) = \frac{1}{2}\left(\boldsymbol{X}_\ell^T\,\boldsymbol{Q}\,\boldsymbol{X}_\ell + \boldsymbol{u}_\ell^T\,\boldsymbol{R}\,\boldsymbol{u}_\ell\right),$$

where $\boldsymbol{Q} \geq 0 \in R^{n\times n}$ and $\boldsymbol{R} > 0 \in R^{m\times m}$ are symmetric time-invariant positive-semi definite and positive definite weighting matrices respectively.

In order to solve the optimal control problem of the system (3), using the performance measure (4), a computational structure or a temporal difference structure is needed. Hence, let the solving value structure $S(\boldsymbol{X}_k)$ be a function of the states $\boldsymbol{X}_k$, then (4) yields

$$S(\boldsymbol{X}_\ell) = \sum_{i=\ell}^{\infty} U(\boldsymbol{X}_i, \boldsymbol{u}_i).$$

The performance index (4) and the cost function U are used to derive Bellman equation for the system (3) so that

$$S(\boldsymbol{X}_\ell) = \frac{1}{2}\left(\boldsymbol{X}_\ell^T\,\boldsymbol{Q}\,\boldsymbol{X}_\ell + \boldsymbol{u}_\ell^T\,\boldsymbol{R}\,\boldsymbol{u}_\ell\right) + S(\boldsymbol{X}_{\ell+1}). \tag{5}$$

The objective of the optimal control problem is to find the optimal value function S^o or equivalently the optimal performance index P^o and the associated optimal policy $\boldsymbol{u}^o$.

Applying the Bellman's optimality principles to (5), in order to find

$$\boldsymbol{A}^{SI} = \frac{\omega_b}{L_{ss}\,L_{rr}\,\delta}\begin{bmatrix} -R_s\,L_{rr} & \alpha_1\,\omega_s & -R_r\,L_m & -\beta_s\,\omega_s \\ -\alpha_1\,\omega_s & -R_s\,L_{rr} & \beta_r\,\omega_s & -R_r\,L_m \\ -R_s\,L_m & \beta_s\,\omega_s & -R_r\,L_{ss} & -\alpha_2\omega_s \\ -\beta_s\omega_s & -R_s\,L_m & \alpha_2\,\omega_s & -R_r\,L_{ss} \end{bmatrix}, \quad \boldsymbol{B}^{SI} = \frac{\omega_b}{L_{ss}\,L_{rr}\,\delta}\begin{bmatrix} -L_{rr} & 0 & L_m & 0 \\ 0 & -L_{rr} & 0 & L_m \\ -L_m & 0 & L_{ss} & 0 \\ 0 & -L_m & 0 & L_{ss} \end{bmatrix}^T,$$

$$\boldsymbol{C}^{SI} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},\ D^{SI} = \boldsymbol{0}_{2\times 4},$$

$$\boldsymbol{A}^{DI} = \frac{\omega_b}{L_{ss}\,\delta_1}\begin{bmatrix} -R_s\,\delta_2 & \gamma_1\,\omega_s & -R_r\,\delta_3 & -\omega_s\,L_m\,\delta_2(1-s) & -R_d\,\delta_4 & -\omega_s\,L_m\,\delta_2\,(1-s) \\ -\gamma_1\,\omega_s & -R_s\,\delta_2 & \omega_s\,L_m\,\delta_2(1-s) & -R_r\,\delta_3 & \omega_s\,L_m\,\delta_2(1-s) & -R_d\delta_4 \\ -R_s\,\delta_3 & \omega_s\,L_{ss}\,\delta_3\,(1-s) & -R_r\,L_{ss}\,X_3 & -\gamma_2\omega_s & R_d\,L_{ss}\,X_2 & -\omega_s\,L_m\,\delta_3\,(1-s) \\ -\omega_s\,L_{ss}\,\delta_3(1-s) & -R_s\delta_3 & \gamma_2\,\omega_s & -R_r\,L_{ss}\,X_3 & \omega_s\,L_m\,\delta_3(1-s) & R_d\,L_{ss}\,X_2 \\ -R_s\,\delta_4 & \omega_s\,L_{ss}\,\delta_4\,(1-s) & R_r\,L_{ss}\,X_2 & -\omega_s\,L_m\,\delta_4\,(1-s) & -R_d\,L_{ss}\,X_1 & -\gamma_3\,\omega_s \\ -\omega_s\,L_{ss}\,\delta_4(1-s) & -R_s\delta_4 & \omega_s\,L_m\,\delta_4(1-s) & R_r\,L_{ss}\,X_2 & \gamma_3\,\omega_s & -R_d\,L_{ss}\,X_1 \end{bmatrix},$$

$$\boldsymbol{B}^{DI} = \frac{\omega_b}{L_{ss}\,\delta_1}\begin{bmatrix} -\delta_2 & 0 & -\delta_3 & 0 & -\delta_4 & 0 \\ 0 & -\delta_2 & 0 & -\delta_3 & 0 & -\delta_4 \\ \delta_3 & 0 & L_{ss}\,X_3 & 0 & -L_{ss}\,X_2 & 0 \\ 0 & \delta_3 & 0 & L_{ss}\,X_3 & 0 & -L_{ss}\,X_2 \end{bmatrix}^T,\ \boldsymbol{C}^{DI} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix},\ \boldsymbol{D}^{DI} = \boldsymbol{0}_{2\times 6}.$$

where $\alpha_1 = L_{ss}\,L_{rr} - s\,L_m^2$, $\alpha_2 = L_m^2 - s\,L_{ss}\,L_{rr}$, $\omega_s = \omega_b/(1-s)$, $\beta_s = L_m\,L_{ss}(1-s)$, $\beta_r = L_m\,L_{rr}(1-s)$, $\delta = 1 - L_m^2/(L_{rr}\,L_{ss})$, $\beta = 1 + L_{rm}/L_s$, $\delta_1 = X_1\,X_3 - X_2^2$, $\delta_2 = L_{rr}\,L_{dd} - \beta^2\,L_m^2$, $\delta_4 = L_m\,L_{rr} - \beta\,L_m^2$, $X_1 = L_{rr} - L_m^2/L_{ss}$, $X_2 = \beta\,L_m - L_m^2/L_{ss}$, $X_3 = L_{dd} - L_m^2/L_{ss}$, $\gamma_1 = L_{ss}\,\delta_2 - s\,L_m(\delta_3 + \delta_4)$, $\gamma_2 = L_m\,\delta_3 + s\,(L_m\,\delta_4 - L_{ss}\,\delta_2)$, and $\gamma_3 = L_m\,\delta_4 + s(L_m\,\delta_3 - L_{ss}\,\delta_2)$.

the optimal policy $\boldsymbol{u}^o$, such that [24]

$$S^o(\boldsymbol{X}_\ell) = \underset{\boldsymbol{u}_i}{\operatorname{argmin}} \sum_{i=\ell}^{\infty} U\left(\boldsymbol{X}_i, \boldsymbol{u}_i\right).$$

Then, the optimal control policy $\boldsymbol{u}^o$ is given by

$$\boldsymbol{u}^o = -\boldsymbol{R}^{-1}\,\boldsymbol{B}^T\,\nabla S^o(\boldsymbol{X}_\ell), \tag{6}$$

where $\nabla S^o(\boldsymbol{X}_\ell) = \partial S^o(\boldsymbol{X}_\ell)/\partial \boldsymbol{X}_\ell$.

The optimal policy (6) works as a mapping $S^o(.) : R^n \rightarrow R^m$ between the states $\boldsymbol{X}_\ell$ and the value function S on one side and the optimal control decisions on the other side.

Remark 1. *The solution of Bellman equation (*5*) depends on finding the optimal policy (*6*), which requires some knowledge about the dynamical model of the wind turbine system (i.e. the optimal policy $\boldsymbol{u}^o$ depends on the matrix $\boldsymbol{B}$).*

3.2 Modified Bellman Equation

Herein, a modified form of Bellman equation (5) is introduced which is necessary to propose the model-free control structure later on. Let the structure of the solving value function depend on both the states and the control policies so that

$$\tilde{S}(\boldsymbol{X}_\ell, \boldsymbol{u}_\ell) = \sum_{i=\ell}^{\infty} U(\boldsymbol{X}_i, \boldsymbol{u}_i). \tag{7}$$

This yields the following modified Bellman equation

$$\tilde{S}(\boldsymbol{X}_\ell, \boldsymbol{u}_\ell) = \frac{1}{2}\left(\boldsymbol{X}_\ell^T \boldsymbol{Q} \boldsymbol{X}_\ell + \boldsymbol{u}_\ell^T \boldsymbol{R} \boldsymbol{u}_\ell\right) + \tilde{S}(\boldsymbol{X}_{\ell+1}, \boldsymbol{u}_{\ell+1}).$$

In order to find the optimal control policy, the Bellman optimality conditions are applied so that

$$\tilde{S}^o(\boldsymbol{X}_\ell, \boldsymbol{u}_\ell) = \underset{\boldsymbol{u}_\ell}{\operatorname{argmin}} \left(U(\boldsymbol{X}_\ell, \boldsymbol{u}_\ell) + \tilde{S}^o(\boldsymbol{X}_{\ell+1}, \boldsymbol{u}_{\ell+1})\right). \tag{8}$$

The derived optimal policy is then expressed as follows

$$\boldsymbol{u}^o = -\boldsymbol{R}^{-1}\,\boldsymbol{B}^T\,\nabla \tilde{S}^o(\boldsymbol{X}_{\ell+1}, \boldsymbol{u}_{\ell+1}), \tag{9}$$

where $\nabla \tilde{S}^o(\boldsymbol{X}_{\ell+1}, \boldsymbol{u}_{\ell+1}) = \partial \tilde{S}^o(\boldsymbol{X}_{\ell+1}, \boldsymbol{u}_{\ell+1})/\partial \boldsymbol{X}_{\ell+1}$.
It is worth to note that, this optimal policy (9) depends on the dynamical model of the wind turbine. The development introduced hereafter handles this challenge.

Using the optimal policy (9) in (8), yields the following modified Bellman optimality equation

$$\tilde{S}^o(\boldsymbol{X}_\ell, \boldsymbol{u}_\ell^o) = U(\boldsymbol{X}_\ell, \boldsymbol{u}_\ell^o) + \tilde{S}^o(\boldsymbol{X}_{\ell+1}, \boldsymbol{u}_{\ell+1}^o). \tag{10}$$

Remark 2. *Solving (*10*) using the optimal policy (*9*) would solve the optimal control problem for the wind turbine system. However, partial knowledge about the dynamics of the DFIG wind turbine is needed in advance. In the following development, a model-free policy approach is proposed to overcome this concern.*

4 Model-Free Policy Iteration Algorithm

In the sequel, a model-free policy approach is followed to solve the dynamic programming problem represented by (10). Then, a policy iteration algorithm is introduced to implement the solution for the optimal control problem of the DFIG-operated wind turbine along with the convergence results.

4.1 Policy Iteration Solution

A solving structure that is different from the one proposed in (5) is given such that

$$C(\boldsymbol{X}_\ell, \boldsymbol{u}_\ell) = \frac{1}{2}[\boldsymbol{X}_\ell^T \;\; \boldsymbol{u}_\ell^T]\;\boldsymbol{M}\;\begin{bmatrix}\boldsymbol{X}_\ell \\ \boldsymbol{u}_\ell\end{bmatrix}, \tag{11}$$

where $\boldsymbol{M} = \begin{bmatrix}\boldsymbol{M}_{XX} & \boldsymbol{M}_{Xu} \\ \boldsymbol{M}_{uX} & \boldsymbol{M}_{uu}\end{bmatrix}$, $M_{\boldsymbol{Xu}}$, $M_{\boldsymbol{uX}}$, $M_{\boldsymbol{XX}}$, and $M_{\boldsymbol{uu}}$ are the states-control, control-states, states-states, and control-control sub-blocks of matrix $\boldsymbol{M}$ respectively.
The dependence of the value structure C on $\boldsymbol{X}_\ell$ and $\boldsymbol{u}_\ell$, facilitates the process of having an optimal model-free control structure as will be shown in the following development.
Applying the Bellman's optimality principles to (11) yields

$$\boldsymbol{u}^o = \underset{\boldsymbol{u}_\ell}{\operatorname{argmin}} \left(C(\boldsymbol{X}_\ell, \boldsymbol{u}_\ell)\right).$$

Then, the model-free optimal policy is given by

$$\boldsymbol{u}^o = -\left[\boldsymbol{M}_{\boldsymbol{u}_\ell \boldsymbol{u}_\ell}^{-1} \;\; \boldsymbol{M}_{\boldsymbol{u}_\ell \boldsymbol{X}_\ell}\right] \;\; \boldsymbol{X}_\ell. \tag{12}$$

The resulting optimal control policy is model-free and in the meantime it is equivalent to the optimal (model-based) policy (9).

Remark 3. *The analogy between the polices (*9*) and (*12*) on one side and the optimal value functions $\tilde{S}^o(\boldsymbol{X}_\ell, \boldsymbol{u}_\ell^o)$ and $C^o(\boldsymbol{X}_\ell, \boldsymbol{u}_\ell^o)$ on another side, results in a novel solution framework structure that does not use any of wind turbine's dynamical parameters. This framework is used to propose the following policy iteration algorithm.*

Algorithm 1 (*Policy Iteration Control Algorithm)*

1. **Initialization:** $C^0(\boldsymbol{X}_0, \boldsymbol{u}_0)$ *and* u_0^0 *(with admissible value)* .
2. **Value evaluation:** $C^r(\dots)$

$$C^r(\boldsymbol{X}_\ell, \boldsymbol{u}_\ell) = \frac{1}{2}(\boldsymbol{X}_\ell^T \boldsymbol{Q} \boldsymbol{X}_\ell + \boldsymbol{u}_\ell^{rT} \boldsymbol{R} \boldsymbol{u}_\ell^r) + C^r(\boldsymbol{X}_{\ell+1}, \boldsymbol{u}_{\ell+1}), \tag{13}$$

where r is the iteration index.

3. **Policy update:**

$$\boldsymbol{u}_\ell^{r+1} = -\left[\boldsymbol{M}_{\boldsymbol{u}_\ell \boldsymbol{u}_\ell}^{-1} \;\; \boldsymbol{M}_{\boldsymbol{u}_\ell \boldsymbol{X}_\ell}\right]^r \;\; \boldsymbol{X}_\ell^r. \tag{14}$$

4. **On convergence of** $\|C^{r+1}(\dots) - C^r(\dots)\|$ end.

Definition 1: The control signal $\boldsymbol{u}_\ell$ is said to be admissible, if it stabilizes system (3).

Remark 4. *The PI-Algorithm 1 defines an online model-free adaptive learning approach to control the DFIG wind turbine system. It uses the value function structure (*11*) and the model-free optimal policy (*12*). This is equivalent to solving the Bellman optimality equation (*10*) using the optimal policy (*9*).*

The policy iteration is an iterative two-step process that solves Bellman equation (10) using the optimal policies (12). The following development studies the convergence properties of performing *Algorithm 1* in real-time.

4.2 Convergence Results

In the sequel, the convergence properties of *Algorithm 1* are studied. It will be shown that, evaluating the value functions $C^r, \forall r$ according to (13) and updating the model-free optimal policies using (14), results in a monotonically decreasing sequence of the value functions $C^r, \forall r$ which, in turn, converges to the optimal solution of (10).

Theorem 1: Let system (3) perform ***Algorithm 1***, where the value function C^r is evaluated using (13) and the policy $\boldsymbol{u}^r$ is updated using (14). If the initial policy $\boldsymbol{u}^0$ is admissible, then

1. The generated policies $\boldsymbol{u}_\ell^{r+1}, \forall r$ are stabilizing and admissible.
2. ***Algorithm 1*** generates a monotonically decreasing sequence of value functions $0 \leq \cdots \leq C^* \leq \cdots \leq C^2 \leq C^1 \leq C^0$, where C^* is the solution to the Bellman optimality equation (10).

Proof: 1. Since $U^r(\boldsymbol{X}_\ell, \boldsymbol{u}_\ell) > 0$, then Bellman equation (13) yields

$$C^r(\boldsymbol{X}_{\ell+1}^r, \boldsymbol{u}_{\ell+1}^r) - C^r(\boldsymbol{X}_\ell, \boldsymbol{u}_\ell) < 0, \forall r. \tag{15}$$

This implies that, the value functions $C^r, \forall r$ are Lyapunov candidates.

The value function C^r given by (11) with the utility function U can be rearranged in the following form.

$$C^r(\boldsymbol{X}_\ell, \boldsymbol{u}_\ell^r) - C^r(\boldsymbol{X}_\ell, \boldsymbol{u}_\ell^{r+1}) = \Delta U(\boldsymbol{u}_\ell^r, \boldsymbol{u}_\ell^{r+1}), \forall r,$$

where $\Delta U(\boldsymbol{u}_\ell^r, \boldsymbol{u}_{\ell+1}^r) = \sum_{i=\ell}^{\infty} \frac{1}{2} \left(\boldsymbol{u}_i^r - \boldsymbol{u}_i^{r+1}\right)^T \boldsymbol{R} \left(\boldsymbol{u}_i^r - \boldsymbol{u}_i^{r+1}\right) + \boldsymbol{u}_i^{(r+1)T} \boldsymbol{R} \left(\boldsymbol{u}_i^r - \boldsymbol{u}_i^{r+1}\right)$.
Since $\Delta U(\boldsymbol{u}_\ell^r, \boldsymbol{u}_{\ell+1}^r) > 0, \forall r, \ell$ then

$$C^r(\boldsymbol{X}_\ell, \boldsymbol{u}_\ell^r) > C^r(\boldsymbol{X}_\ell, \boldsymbol{u}_\ell^{r+1}), \ \forall r.$$

Thus $\boldsymbol{u}_\ell^{r+1}, \forall r, \ell$ are stabilizing policies and hence admissible, if the initial policy $\boldsymbol{u}_\ell^0$ is admissible. This is because, the respective value functions, due to these policies, are decreasing and bounded below by 0.

2. Using the policy $\boldsymbol{u}_\ell^{r+1}, \forall r$ in Bellman equation (13) yields

$$C^r(\boldsymbol{X}_{\ell+1}^{r+1}, \boldsymbol{u}_{\ell+1}^{r+1}) - C^r(\boldsymbol{X}_\ell, \boldsymbol{u}_\ell^{r+1}) < -U(\boldsymbol{X}_\ell, \boldsymbol{u}_\ell^{r+1}) < 0, \tag{16}$$

and

$$C^{r+1}(\boldsymbol{X}_{\ell+1}^{r+1}, \boldsymbol{u}_{\ell+1}^{r+1}) - C^{r+1}(\boldsymbol{X}_\ell, \boldsymbol{u}_\ell^{r+1}) + U(\boldsymbol{X}_\ell, \boldsymbol{u}_\ell^{r+1}) = 0, \forall r. \tag{17}$$

Then (16) and (17) result in the following inequality

$$C^r(\boldsymbol{X}_{\ell+1}^{r+1}, \boldsymbol{u}_{\ell+1}^{r+1}) - C^r(\boldsymbol{X}_\ell, \boldsymbol{u}_\ell^{r+1}) < C^{r+1}(\boldsymbol{X}_{\ell+1}^{r+1}, \boldsymbol{u}_{\ell+1}^{r+1}) - C^{r+1}(\boldsymbol{X}_\ell, \boldsymbol{u}_\ell^{r+1}). \tag{18}$$

Taking the infinite summation (over the time-index ℓ) on the inequality (18) yields

$$\sum_{\ell=L}^{\infty} \left(C^r(\boldsymbol{X}_{\ell+1}^{r+1}, \boldsymbol{u}_{\ell+1}^{r+1}) - C^r(\boldsymbol{X}_\ell, \boldsymbol{u}_\ell^{r+1})\right) < \sum_{\ell=L}^{\infty} \left(C^{r+1}(\boldsymbol{X}_{\ell+1}^{r+1}, \boldsymbol{u}_{\ell+1}^{r+1}) - C^{r+1}(\boldsymbol{X}_\ell, \boldsymbol{u}_\ell^{r+1})\right).$$

Therefore,

$$C^r(\boldsymbol{X}_\infty^{r+1}, \boldsymbol{u}_\infty^{r+1}) - C^r(\boldsymbol{X}_L, \boldsymbol{u}_L^{r+1}) < C^{r+1}(\boldsymbol{X}_\infty^{r+1}, \boldsymbol{u}_\infty^{r+1}) - C^{r+1}(\boldsymbol{X}_L, \boldsymbol{u}_L^{r+1}).$$

The asymptotic stability properties of part *(1)* yield

$$C^r(\boldsymbol{X}_\infty^{r+1}, \boldsymbol{u}_\infty^{r+1}) = C^{r+1}(\boldsymbol{X}_\infty^{r+1}, \boldsymbol{u}_\infty^{r+1}) = 0$$

Then, $C^r(\boldsymbol{X}_L, \boldsymbol{u}_L^{r+1}) > C^{r+1}(\boldsymbol{X}_L, \boldsymbol{u}_L^{r+1})$. Therefore by induction, this inequality yields

$$0 < \cdots < C^{r+1} < C^r < \cdots < C^2 < C^1 < C^0. \tag{19}$$

This inequality reveals that, the ***Algorithm 1*** generates a sequence of monotonically decreasing value functions $C^r, \forall r$. The maximum bound for this sequence is C^0 and its lower bound is *0*. Therefore, the decreasing sequence (19) is bounded below and converges to the optimal value function C^* such that

$$0 < \cdots < C^* < \cdots < C^2 < C^1 < C^0. \tag{20}$$

The value function C^* is the optimal solution for the Bellman optimality equation (10).

The ***Policy Iteration Algorithm 1*** is implemented in real-time using means of the adaptive critics and neural network structures as will be shown in the following section.

5 Adaptive Critics Implementation

In the sequel, the adaptive critics are used to implement the online policy iteration solution. The optimal policy (12) and the value function (11) are approximated using the actor-critic neural network structures. The critic neural structure approximates the value function (11), while the actor neural structure approximates the model-free optimal policy (12). This scheme solves Bellman optimality equation (10) in real-time. In addition, only one layer of neural networks is used, while the weights are tuned using gradient descent techniques.

5.1 Adaptive Critics

The adaptive critics are adopted to implement ***Algorithm 1***. Two neural network structures are used to approximate the value function C and the policy u as shown in Figure 2. The actor-critic neural network structures are tuned in real-time using the data measured along the DFIG wind turbine's trajectory. The following development shows the training processes for the actor and critic neural networks. The actor structure approximates the optimal strategies or decisions that minimize certain cost function, while the critic structure approximates the optimal value function. This process requires an initial admissible policy as illustrated by ***Theorem 1***.

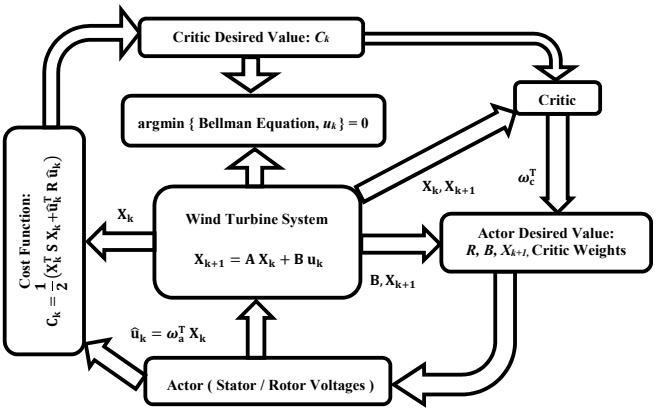

Fig. 2: The adaptive critics' system design.

5.2 Online Actor-Critic Tuning

The following setup aims to develop the online training rules for the actor and critic neural network weights, which is equivalent to performing ***Algorithm 1*** and solving the Bellman optimality equation (10).
The actor structure approximates the policy (14) such that

$$\hat{\boldsymbol{u}}_\ell = \boldsymbol{\omega}_a^T \boldsymbol{X}_\ell,$$

where $\boldsymbol{\omega}_a^T \in R^{n\times m}$ are the actor weights.
The critic structure approximates the value function (11) such that

$$\hat{C}(\boldsymbol{X}_\ell, \hat{\boldsymbol{u}}_\ell) = \frac{1}{2}\boldsymbol{Z}_\ell^T\, \boldsymbol{\omega}_c^T\, \boldsymbol{Z}_\ell, \tag{21}$$

where $\boldsymbol{\omega}_c^T \in R^{(n+m)\times(n+m)}$ are the critic weights, $\hat{\boldsymbol{u}}_\ell$ is the approximation of the model-free policy, and $\boldsymbol{Z}_\ell^T = [\boldsymbol{X}_\ell^T \;\; \hat{\boldsymbol{u}}_\ell^T]$.
The desired control strategy is given by

$$\boldsymbol{u}_\ell^{desired} = -\left[\boldsymbol{\omega}_{c\hat{\boldsymbol{u}}_\ell\hat{\boldsymbol{u}}_\ell}^{-1} \;\; \boldsymbol{\omega}_{c\hat{\boldsymbol{u}}_\ell\boldsymbol{X}_\ell}\right] \;\; \boldsymbol{X}_\ell \tag{22}$$

The actor's approximation error is

$$\boldsymbol{\eta}_a = \hat{\boldsymbol{u}}_\ell - \boldsymbol{u}_\ell^{desired}.$$

Applying the gradient descent technique, yields the actor weights' update rule such that

$$\boldsymbol{\omega}_a^{(r+1)T} = \boldsymbol{\omega}_a^{(r)T} - \gamma_a \left(\hat{\boldsymbol{u}}_\ell^{(r)} - \boldsymbol{u}_\ell^{(r)desired}\right) \boldsymbol{X}_\ell^T. \tag{23}$$

where $0 < \gamma_a < 1$ is the actor's learning rate.
The value function approximation (21) is rearranged in a form suitable to perform the policy iteration ***Algorithm 1*** such that

$$\hat{C}(\boldsymbol{X}_\ell, \hat{\boldsymbol{u}}_\ell) = \tilde{\boldsymbol{\omega}}_c^T\, \tilde{\boldsymbol{Z}}_\ell,$$

where $\tilde{\boldsymbol{\omega}}_c$ is a vector transformation of the symmetric square matrix $\boldsymbol{\omega}_c$ with $(n+m)\times(n+m+1)/2$ instances and $\tilde{\boldsymbol{Z}}$ is the respective combined state-control vector.
The approximation $\tilde{C}_{\ell,\ell+1} = \hat{C}(\boldsymbol{X}_\ell, \hat{\boldsymbol{u}}_\ell) - \hat{C}(\boldsymbol{X}_{\ell+1}, \hat{\boldsymbol{u}}_{\ell+1})$ is represented by

$$\tilde{C}_{\ell,\ell+1} = \hat{C}(\boldsymbol{X}_\ell, \hat{\boldsymbol{u}}_\ell) - \hat{C}(\boldsymbol{X}_{\ell+1}, \hat{\boldsymbol{u}}_{\ell+1}) = \tilde{\boldsymbol{\omega}}_c^T \left(\tilde{\boldsymbol{Z}}_\ell - \tilde{\boldsymbol{Z}}_{\ell+1}\right). \tag{24}$$

The desired value function is given by

$$C_l^{desired} = \frac{1}{2}\left(\boldsymbol{X}_\ell^T \boldsymbol{Q} \boldsymbol{X}_\ell + \hat{\boldsymbol{u}}_\ell^T \boldsymbol{R} \hat{\boldsymbol{u}}_\ell\right),$$

The critic's approximation error is

$$\eta_c = \tilde{C}_{\ell,\ell+1} - C_l^{desired}. \tag{25}$$

Applying the gradient descent technique, yields the critic weights' update rule such that

$$\tilde{\boldsymbol{\omega}}_c^{(r+1)T} = \tilde{\boldsymbol{\omega}}_c^{(r)T} - \gamma_c \left(\tilde{\boldsymbol{\omega}}_c^{(r)T} F^{sm}\left(\tilde{\boldsymbol{Z}}_\ell - \tilde{\boldsymbol{Z}}_{\ell+1}\right) - F^{rv}\left(C_\ell^{Target}\right)\right) F^{sm}\left(\tilde{\boldsymbol{Z}}_\ell - \tilde{\boldsymbol{Z}}_{\ell+1}\right)^T. \tag{26}$$

where $0 < \gamma_c < 1$ is the critic's learning rate, F^{rv} is a row vector of size $(n+m)\times(n+m+1)/2$, and F^{sm} is a square matrix of size $(n+m)\times(n+m+1)/2$.

The following algorithm summarizes the online actor-critic tuning process to implement the solution for ***Algorithm 1***.

Algorithm 2 (*Online Actor-Critic Tuning*)

1. **Initialization:** *Initialize the critic and actor weights* $\boldsymbol{\omega}_c^0$ *and* $\boldsymbol{\omega}_a^0$.
2. **Outer-process (for *j* instances) Initialize:** $\boldsymbol{X}^0$
3. **Inside-process (for** $t = (n+m)\times(n+m+1)/2$ **iterations)**
   (a) **Transfer weights:** $\boldsymbol{\omega}_c^t = \boldsymbol{\omega}_c^j, \boldsymbol{\omega}_a^t = \boldsymbol{\omega}_a^j$.
   (b) **Evaluate the policy:** $\hat{\boldsymbol{u}}_\ell^t$ using (22).
   (c) **Evaluate the dynamics:** $\hat{\boldsymbol{X}}_{(\ell+1)}^t$.
   (d) **Evaluate the value function:** $\tilde{C}_{\ell,\ell+1}$ using (24).
4. **Update the value:** Critic weights update using (26).
5. **Update the policy:** Actor weights update using (23).
6. **Check the convergence:** $\|\hat{C}^{j+1}(\dots) - \hat{C}^j(\dots)\|$ End.

6 Simulation and Stability Results

Three simulated case studies are conducted to test the validity of the developed model-free control structure and highlight the advantages of its closed-loop characteristics. The controller does not require any prior knowledge of the wind-turbine's dynamics. Two wind turbine models are considered: single and double-cage induction generators, referred to as SCIG and DCIG, respectively, for short. The dynamical models of both turbines are given in (1) and (2), respectively. For simulation purpose, the state space equations were converted to the discrete-time domain, in the form (3). The turbine's parameters, which are used in the simulation, are provided in Table 1.

Table 1 Wind Turbine Parameters [2, 10]

Parameter	Value	Parameter	Value
ω_b	1.01 $p.u.$	s	-0.005
R_s	0.00488 $p.u.$	R_r	0.00549 $p.u.$
R_d	0.2696 $p.u.$	L_{rm}	0.02 $p.u.$
L_{ss}	0.09241 $p.u.$	L_{rr}	0.09955 $p.u.$
L_m	3.95279 $p.u.$	J	3.5 s

The simulations were run in Matlab 2017 on a server with 16 virtual CPUs and 48 GB of memory. The sampling period was taken as 0.01 s.

The matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ for the SCIG plant are given by

$$\boldsymbol{A} = \begin{bmatrix} 1.0000 & -0.0001 & 0.0000 & 0.0003 \\ 0.0001 & 1.0000 & -0.0003 & 0.0000 \\ 0.0000 & -0.0002 & 0.9999 & 0.0102 \\ 0.0002 & 0.0000 & -0.0102 & 0.9999 \end{bmatrix},$$

$$\boldsymbol{B} = \begin{bmatrix} 0.0001 & 0.0000 & -0.0026 & 0.0000 \\ -0.0000 & 0.0001 & -0.0000 & -0.0026 \\ 0.0026 & 0.0000 & -0.0001 & 0.0000 \\ -0.0000 & 0.0026 & -0.0000 & -0.0001 \end{bmatrix}.$$

The matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ for the DCIG plant are taken as

$$\boldsymbol{A} = \begin{bmatrix} 1.0000 & -0.0002 & -0.0000 & 0.0053 & 0.0003 & 0.0053 \\ 0.0002 & 1.0000 & -0.0053 & -0.0000 & -0.0053 & 0.0003 \\ 0.0000 & -0.0001 & 1.0000 & 0.0051 & -0.0004 & 0.0052 \\ 0.0001 & 0.0000 & -0.0051 & 1.0000 & -0.0052 & -0.0004 \\ 0.0000 & -0.0001 & -0.0000 & 0.0051 & 1.0003 & 0.0051 \\ 0.0001 & 0.0000 & -0.0051 & -0.0000 & -0.0051 & 1.0003 \end{bmatrix},$$

$$\boldsymbol{B} = \begin{bmatrix} 0.0013 & 0.0000 & -0.0013 & 0.0000 \\ -0.0000 & 0.0013 & -0.0000 & -0.0013 \\ 0.0013 & 0.0000 & -0.0013 & 0.0000 \\ -0.0000 & 0.0013 & -0.0000 & -0.0013 \\ 0.0013 & 0.0000 & 0.0013 & -0.0000 \\ -0.0000 & 0.0013 & 0.0000 & 0.0013 \end{bmatrix}.$$

These parameters lead to unstable open-loop turbine models, as shown by their open-loop poles in Table 2. In the following simulated case studies, the system poles are illustrated graphically such that, open-loop poles are shown in blue 'o'; closed-loop poles as they evolve during the learning process are printed in green '*'; the final closed-loop poles after learning are marked by blue '+'; and the closed-loop poles as solved by the Riccati approach are illustrated by red '+'. The latter is only relevant in one of the simulation cases, as shall be later explained. The actor-critic neural network parameters for both turbines were set to the identity matrix for $\boldsymbol{Q}$ and $\boldsymbol{R}$, and $\gamma_a = \gamma_c = 0.01$.

6.1 *First case study: nominal model*

Although the intelligent control algorithm does not assume any knowledge of the turbine dynamics, equations (1) and (2) were

Table 2 Open- and Closed-Loop Poles

	SCIG	DCIG
Open loop	$1.0000\, e^{\pm j0.0001}$ $0.9999\, e^{\pm j0.0102}$	$1.0004\, e^{\pm j0.0001}$ $1.0001\, e^{\pm j0.0101}$ $1.0000\, e^{\pm j0.0001}$
Nominal model	$0.9943\, e^{\pm j0.0096}$ $0.9944,\ 0.9904$	$0.9994\, e^{\pm j0.0009}$ $0.9964\, e^{\pm j0.0087}$ $0.9995,\ 0.9978$
Gaussian noise	$0.9933\, e^{\pm j0.0027}$ $0.9929\, e^{\pm j0.0090}$	$0.9974\, e^{\pm j0.0112}$ $0.9963,\ 0.9986,\ 0.9993,\ 0.9998$
(Riccati)	$0.9975\, e^{\pm j0.0101}$ $0.9974\, e^{\pm j0.0001}$	$0.9996\, e^{\pm j0.0001}$ $0.9978\, e^{\pm j0.0101}$ $0.9978\, e^{\pm j0.0001}$
Coexisting situations	$0.9985\, e^{\pm j0.0083}$ $0.9935,\ 0.9990$	$0.9998\, e^{\pm j0.0008}$ $0.9990\, e^{\pm j0.0056}$ $0.9944,\ 0.9991$

adopted to simulate the nominal dynamical model of the wind turbines. The results of this case are shown in Figures 3, 4, and 5. The weights of the actor and critic neural networks in Figure 5 are not shown for the whole simulation time. This is because, after some learning time, their variations become no longer significant to be visually noticeable.

6.2 Second case study: Gaussian modeling noise

In this case study, the intelligent control scheme is tested against a radically changing uncertainty in the plant dynamics. To simulate such an unstructured uncertainty, an unknown time-dependent Gaussian noise is injected into the turbine dynamics (1) and (2) to make it in the form of $\boldsymbol{X}_{k+1} = (\boldsymbol{A} + \Delta\boldsymbol{A}_k)\,\boldsymbol{X}_k + \left(\boldsymbol{B}\,\boldsymbol{u}_k^R + \Delta\boldsymbol{B}_k\,\boldsymbol{u}_k\right)$, where $\Delta\boldsymbol{A}_k$ and $\Delta\boldsymbol{B}_k$ are the added noise in the system and input matrices at time instance k. For each matrix, the noise is sampled from a normal distribution $\mathcal{N}(0,\ 1)$, which brings up the uncertainty to $\pm 50\%$ of the matrix nominal value. The term $\boldsymbol{u}^R$ is the state feedback control law obtained by solving the Riccati equation applied to the nominal matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ and the weight matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$. The remaining control signal $\boldsymbol{u}$ is generated by the intelligent control approach to introduce more robustness in the face of the aggressively varying plant dynamics. The simulation results of this experiment are reported in Figures 6, 7, and 8.

6.3 Third case study: challenging coexisting situations

The performance of the controller is further validated by monitoring the wind turbine operation under ultimately challenging and combined situations, such as severe wind changes with load excursions (on and off loads) and abrupt variations in the load levels. These events are simulated by introducing time varying disturbances in the respective currents (states) and voltages (control signals) for the single and double cage induction generators. The disturbance levels are set up to 0.5 per unit for each current variable and 0.2 per unit for each voltage control signal. The performance results are revealed in Figures 9, 10, and 11.

6.4 Discussion

The proposed controller was able to stabilize the closed-loop system in all three cases, as reported in Table 2. The stabilization process took longer with the DCIG than with its single-cage counter part. This is expected since the former has more complex dynamics. As a result, the actor-critic neural networks have more weights to adapt for learning the DCIG than the SCIG model. This also explains why the controller required more time to compensate for the time-varying modeling uncertainties in the last two case studies.

The controller's most challenging case was to regulate the DCIG with the Gaussian modeling uncertainty. This is clear from the largest search space it had to exploit for the closed-loop poles, as shown by the area scattered in green in Figure 7(b). In this scenario, the search expanded towards the unstable region (outside the unit circle) before converging to the asymptotic stability region. It is interesting to notice that the dominant closed-loop poles in this case study with the SCIG are not only faster than those obtained for the nominal case but also faster than those obtained by the classical Riccati control approach. This is clearly demonstrated in Figure 7(a).

In spite of the challenging random abrupt situations imposed on the operation of the wind turbines in the third case study, the model-free controller was able to learn the system's dynamical behavior and shift the closed-loop poles towards the region of asymptotic stability. After a relatively fluctuating transient phase, an equilibrium point was eventually reached. This extreme scenario posed a real challenge to the controller. Not only can this be seen by the relatively longer convergence time but also by observing the dominant closed-loop poles which are barely inside the unit circle and so led to a relatively slower convergence rate.

Remark 5. *The grid faults cause variations in the coupled stator-rotor dynamics of the DFIG-based wind turbine. This is when the crowbar system activates the protection mechanism. The simplified models of the crowbar systems are satisfactory representatives of the real systems when the resistance values are small (i.e. when the dc clamp effect does not occur) [28]. However, with high resistance values, the Low Voltage Ride-Through (LVRT) operation is severely degraded and these models are no longer effective. Thus, the modeling part plays a crucial role in the design and performance of the protection mechanism. The developed intelligent controller overcomes two related concerns. First, it is able to intelligently incorporate the uncertainties in the dynamics of the DFIG-based wind turbine which involve the grid faults. Second, it is able to reject disturbances in the rotor voltages. The proposed controller provides a natural support during these situations, since the variations in the dynamical terms, due to the grid faults (i.e. $\Delta\boldsymbol{A}_k$), are implicitly embedded in the coupled rotor-stator dynamics [28]. This is clearly demonstrated in the second simulation scenario, which showed the controller's capability to deal with large deviations in the stator-rotor coupled dynamics. It is also illustrated in the third scenario, which revealed how the controller could support the protection system when the rotor voltages are disturbed (i.e. $\Delta\boldsymbol{u}_k$).*

7 Conclusion

The paper introduces a new mathematical setup and its computational mechanism to define a real-time model-free optimal controller for the DFIG-based wind turbines. The online intelligent controller employs a novel Reinforcement Learning approach based on a policy iteration process without the need for any prior knowledge of the DFIG-based wind turbine dynamics. The online control algorithm is implemented using single-layer multi-perceptron actor-critic neural networks with gradient descent tuning laws. This flexible control mechanism provides a high degree of robustness for several types of stochastic nonlinear plants, such as wind renewable energy schemes with unpredictable wind conditions, machine dynamics, induction generator interactions with electrical grid interfaces, and static-multi rotor induction generator saturation and parametric variations. The controller's online closed-loop characteristics revealed its ability to asymptotically stabilize the wind turbine under ultimately disturbing scenarios. Moreover, it outperformed the classical Riccati control solution in terms of closed-loop stability and robustness against uncertainties, despite the latter's dependence on a perfectly known dynamical model. The proposed framework can be extended to optimize a mix of coordinated/scheduled energy management of smartgrid/hybrid smartgrids with storage units including Photovoltaic, Wind, Fuel Cell, Li Ion, tidal/wave, and micro gas backup turbines.

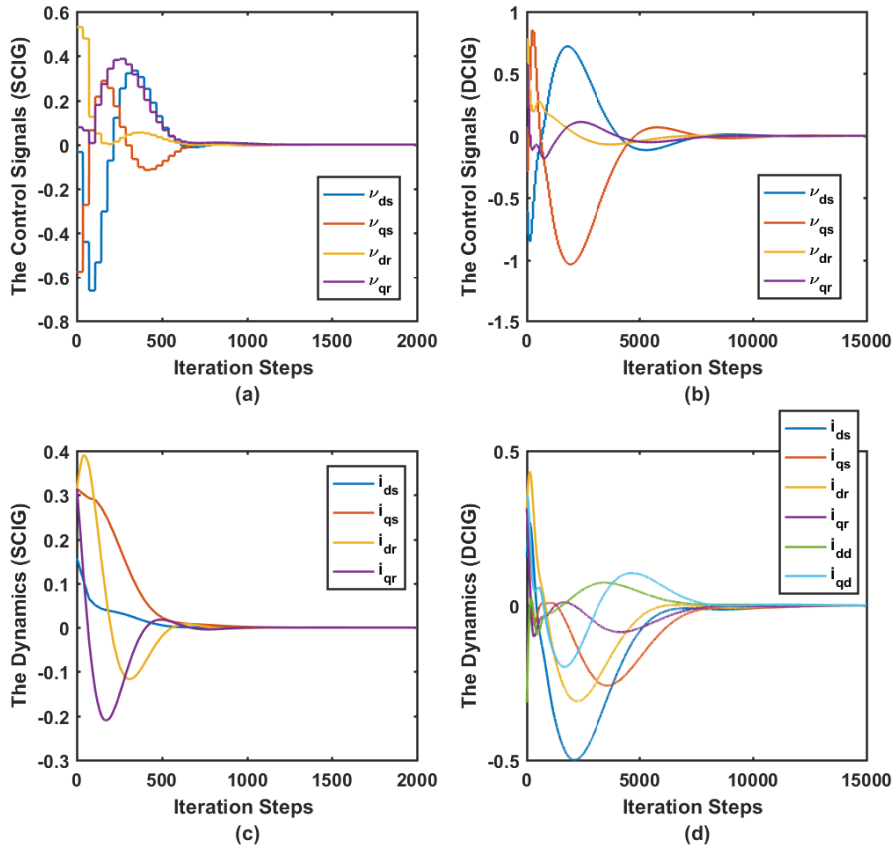

Fig. 3: Dynamical results of the first case study (nominal model): (a) The control signals for the SCIG turbine. (b) The control signals for the DCIG turbine. (c) The dynamics for the SCIG turbine. (d) The dynamics for the DCIG turbine.

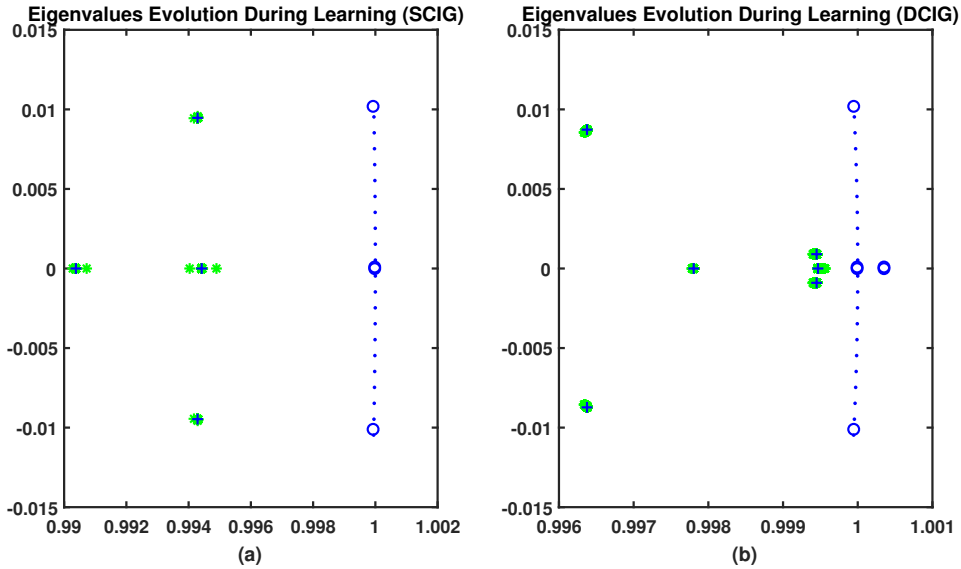

Fig. 4: System poles of the first case study (nominal model): (a) The system poles for the SCIG turbine. (b) The system poles for the DCIG turbine.

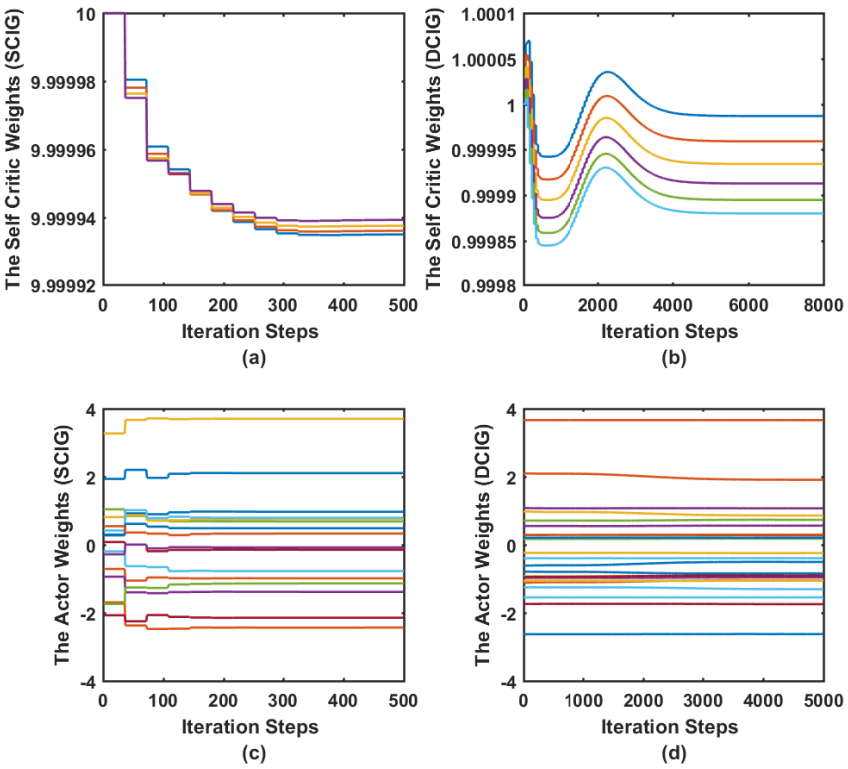

Fig. 5: Actor and critic weights adaptation in the first case study (nominal model): (a) The critic weights for the SCIG turbine. (b) The critic weights for the DCIG turbine. (c) The actor weights for the SCIG turbine. (d) The actor weights for the DCIG turbine.

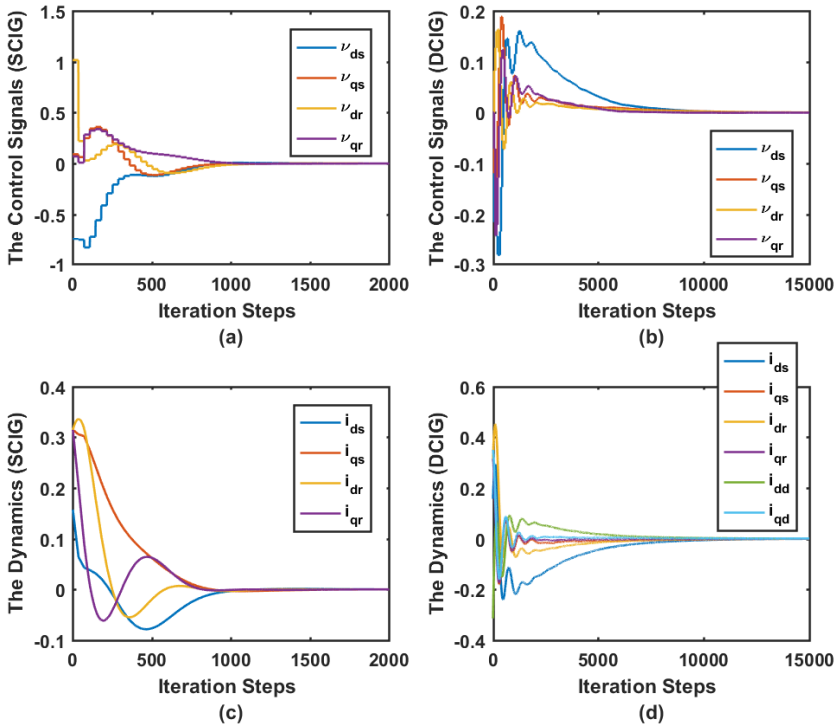

Fig. 6: Dynamical results of the second case study (Gaussian noise): (a) The control signals for the SCIG turbine. (b) The control signals for the DCIG turbine. (c) The dynamics for the SCIG turbine. (d) The dynamics for the DCIG turbine.

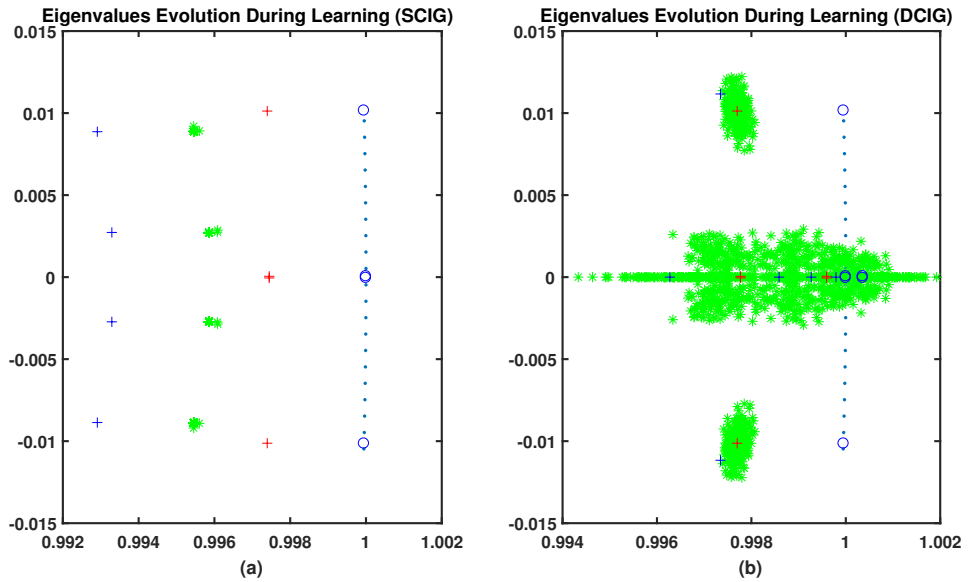

Fig. 7: System poles of the second case study (Gaussian noise): (a) The system poles for the SCIG turbine. (b) The system poles for the DCIG turbine.

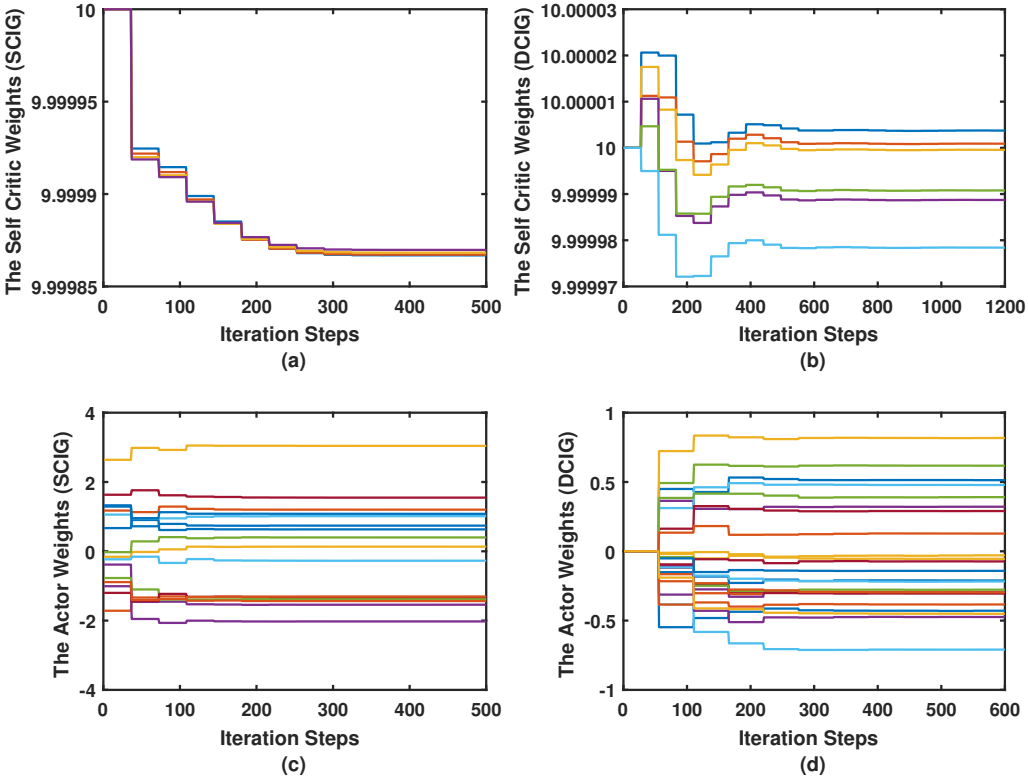

Fig. 8: Actor and critic weights adaptation in the second case study (Gaussian noise):(a) The critic weights for the SCIG turbine. (b) The critic weights for the DCIG turbine. (c) The actor weights for the SCIG turbine. (d) The actor weights for the DCIG turbine.

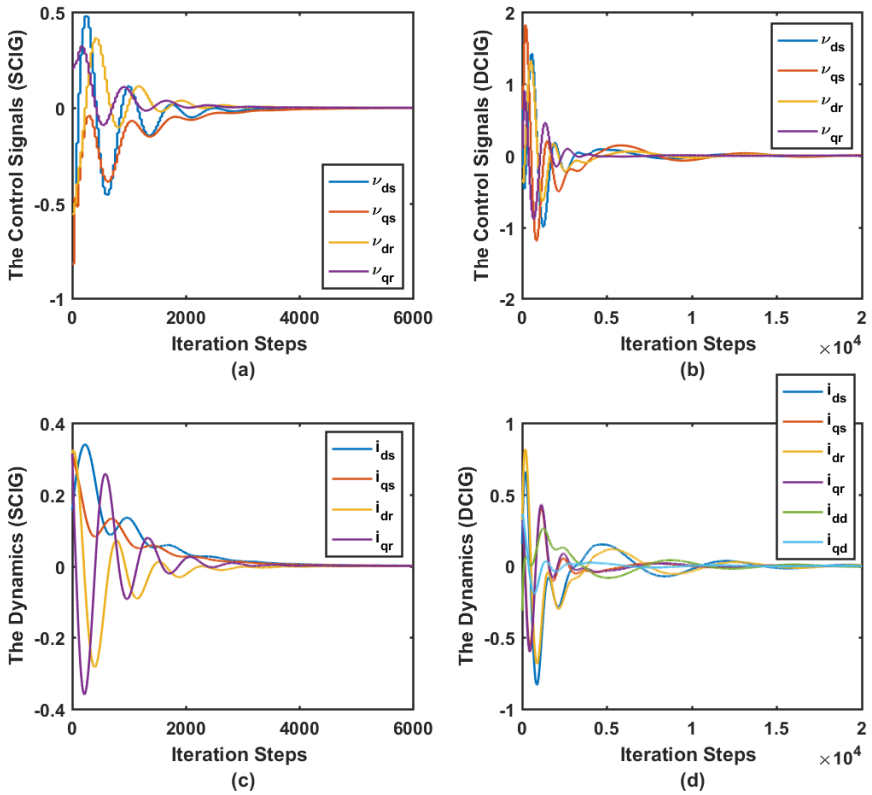

Fig. 9: Dynamical results of the third case study (coexisting situations):(a) The control signals for the SCIG turbine. (b) The control signals for the DCIG turbine. (c) The dynamics for the SCIG turbine. (d) The dynamics for the DCIG turbine.

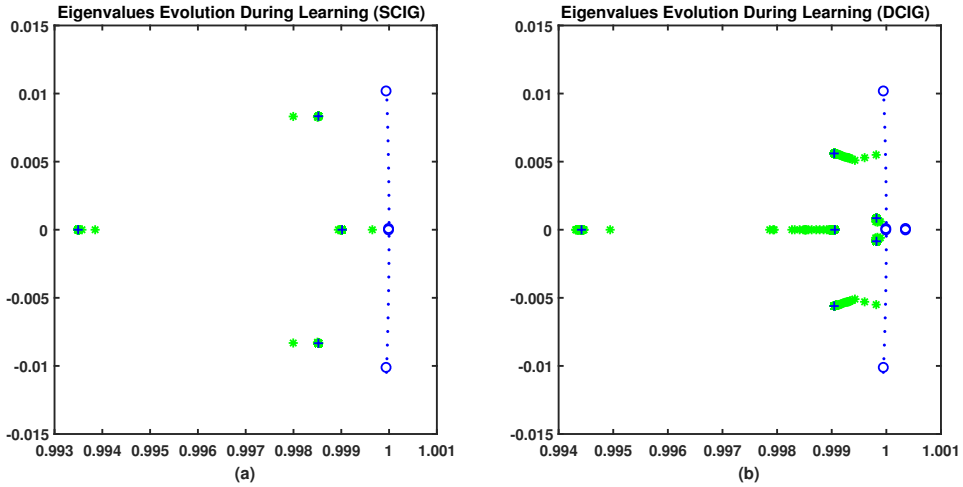

Fig. 10: System poles of the third case study (coexisting situations): (a) The system poles for the SCIG turbine. (b) The system poles for the DCIG turbine.

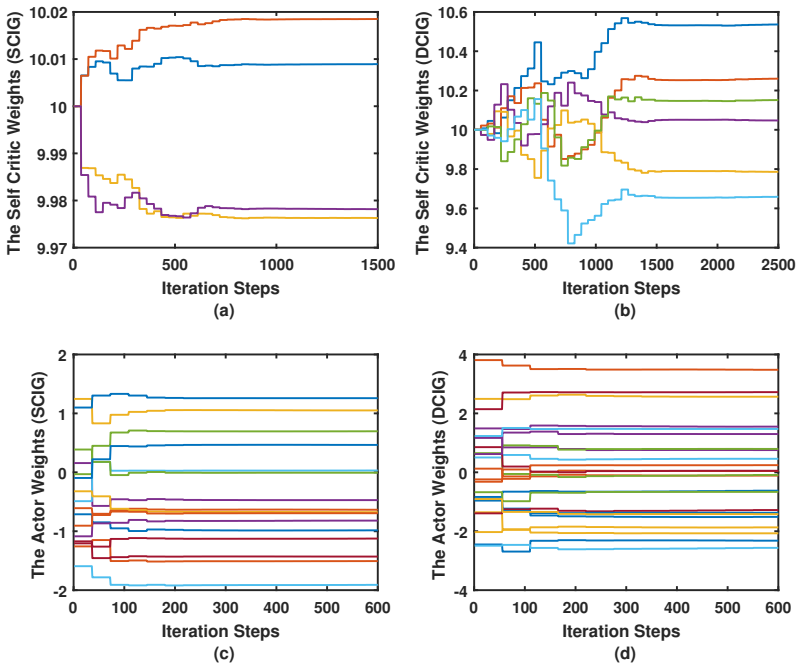

Fig. 11: Actor and critic weights adaptation in the third case study (coexisting situations): (a) The critic weights for the SCIG turbine. (b) The critic weights for the DCIG turbine. (c) The actor weights for the SCIG turbine. (d) The actor weights for the DCIG turbine.

8 References

1 Kazachkov, Y., Feltes, J., Zavadil, R.: 'Modeling wind farms for power system stability studies'. Proc. IEEE Pow. Eng. Soc. Gen. Meet., Toronto, Canada, July 2003, pp. 15261–1533

2 Ekanayake, J., Holdsworth, L., Wu, X., Jenkins, N.: 'Dynamic modeling of doubly fed induction generator wind turbines', IEEE Trans. on Pow. Sys., 2003, 18, (2), pp. 803–809

3 Muller, S., Deicke, M., De Doncker, R.: 'Doubly fed induction generator systems for wind turbines', IEEE Ind. Applic. Mag., 2002, 8, (3), pp. 26–33

4 Lei, Y., Mullane, A., Lightbody, G., Yacamini, R.: 'Modeling of the Wind Turbine with a Doubly Fed Induction Generator for Grid Integration Studies', IEEE Trans. on Ener. Conv., 2006, 21, (1), pp. 257–264

5 Anaya-Lara, O., Jenkins, N., Ekanayake, J., Cartwright, P., Hughes, M.: 'Wind Energy Generation Modelling and Control'(Wiley, 2009, 1st edn.)

6 Akhmatov, V., Nielsen, A., Pedersen, J., Nymann, O.: 'Variable speed wind turbines with multi-pole synchronous permanent magnet generators', Wind Eng., 2003, 27, (6), pp. 531–548

7 Ramtharan, G., Jenkins, N. Anaya-Lara, O.: 'Modelling and control of synchronous generators for wide-range variable-speed wind turbines', Wind Ener., 2007, 10, (3), pp. 231–246

8 Slootweg, J., Polinder, H., Kling, W.: 'Dynamic modeling of a wind turbine with doubly fed induction generator', Proc. IEEE Powr. Eng. Soc. Summer Meet., Vancouver, Canada, July 2001, pp. 644–649

9 Feijoo, A., Cidras, J., Carrillo, C.: 'A third order model for the doubly-fed induction machine', Elect. Pow. Syst. Res., 2000, 56, (2), pp. 121–127

10 Ugalde-Loo, C., Ekanayake, J., Jenkins, N.: 'State-Space Modeling of Wind Turbine Generators for Power System Studies', IEEE Trans. on Ind. Appl., 2013, 49, (1), pp. 223–232

11 Slootweg, J., De Haan, S., Polinder, H., Kling, W.: 'General model for representing variable speed wind turbines in power system dynamics simulations', IEEE Trans. Pow. Sys., 2003, 18, (1), pp. 144–151

12 Li, H., Chen, Z.:'Overview of different wind generator systems and their comparisons', IET Ren. Pow. Gen., 2008, 2, (2), pp. 123–138

13 Ramtharan, G., Jenkins, N., Ekanayake, J.:'Frequency support from doubly fed induction generator wind turbines', IET Ren. Pow. Gen., 2007, 1, (1), pp. 3–9

14 Melicio, R., Mendes, V., Catalao, J.: 'Fractional-order control and simulation of wind energy systems with PMSG/full-power converter topology', Ener. Conver. and Manag., 2010, 51, (6), pp. 1250–1258

15 Moafi, M., Marzband, M., Savaghebi, M., Guerrero, J.: 'Energy management system based on fuzzy fractional order PID controller for transient stability improvement in microgrids with energy storage', Int. Trans. Elec. Ener. Sys., 2016, 26, (10), pp. 2087–2106

16 Xu, L., Cartwright, P.: 'Direct active and reactive power control of DFIG for wind energy generation', IEEE Trans. on Ener. Conv., 2006, 21, (3), pp. 750–758

17 Xu, L., Wang, Y.: 'Dynamic Modeling and Control of DFIG-Based Wind Turbines Under Unbalanced Network Conditions', IEEE Trans. on Pow. Sys., 2007, 22, (1), pp. 314-323

18 Hughes, F., Anaya-Lara, O., Jenkins, N., Strbac, G.: 'Control of DFIG-based wind generation for power network support', IEEE Trans. on Pow. Sys., 2005, 20, (4), pp. 1958–1966

19 Howard, R.: 'Dynamic Programming and Markov Processes', (MIT Press, 1960, 1ST EDN.)

20 Werbos, P.: 'Approximate dynamic programming for real-time control and neural modeling', 'Hand. of intel. con.: Neur., fuz., and adap. appr.', (VAN NOSTRAND REINHOLD, 1992,1st edn.), pp. 493–525

21 Bertsekas, D., Tsitsiklis, J.: 'Neuro-dynamic programming: an overview', IEEE Proc. Dec. and Con., New Orleans, USA, December 1995, pp. 560–564

22 Werbos, P.: 'A menu of designs for reinforcement learning over time', Neur. net. for con., 1990, pp. 67–95

23 Barto, A.: 'Reinforcement learning: An introduction', (MIT Press, 1998, 1st edn.)

24 Lewis, F., Vrabie, D., Syrmos., V.: 'Optimal Control', (John Wiley, 2012, 3rd edn.)

25 Abouheaf, M., Mahmoud, M., Hussain, S.: 'A Novel Approach to Control of Autonomous Microgrid Systems', Int. Jour. of Ener. Eng., 2015, 5, (5), pp. 125–136

26 Alyazidi, N., Mahmoud, M., Abouheaf, M.: 'Adaptive critics based cooperative control scheme for islanded Microgrids', Neurocomput., 2018, 272, pp. 532–541

27 Zhang, Z., Sun, Y., Lin, J., Li, G.:'Coordinated frequency regulation by doubly fed induction generator-based wind power plants', IET Ren. Pow. Gen., 2012, 6, (1), pp. 38–47

28 Zhou, H., Yang, G., and Li, D.:'Short circuit current analysis of DFIG wind turbines with crowbar protection', Int. Conf. on Elec. Mach. and Sys., Tokyo, Japan, Nov 2009, pp. 1–6

29 Xiang, D., Ran, L., Tavner, P., and Yang, S.: 'Control of a doubly fed induction generator in a wind turbine during grid fault ride-through', IEEE Trans. on Ener. Conv., 2006, 21, (3), pp. 652–662

30 Lopez, J., Sanchis, P., Roboam, X., and Marroyo, L.: 'Dynamic Behavior of the Doubly Fed Induction Generator During Three-Phase Voltage Dips', IEEE Trans. on Ener. Conv., 2007, 22, (3), pp. 709–717

31 Johan Morren, Sjoerd W. H. de Haan, 'Short Circuit Current of Wind Turbines with Doubly Fed Induction Generator', IEEE Trans. on Ener. Conv., 2007, 22, (1), pp. 174–180