Heliyon 10 (2024) e35712

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Robust control of a wind energy conversion system: FPGA real-time implementation

Abdelhafid El Attafi [a], Houda El Alami [a], Badre Bossoufi [a,*], Dokhyl AlQahtani [b], Saad Motahhir [c], Mishari Metab Almalki [d], Thamer A.H. Alghamdi [e,f]

[a] LIMAS Laboratory, Faculty of Sciences Dhar El Mahraz, Sidi Mohammed Ben Abdellah University, Fez, 30003, Morocco
[b] Department of Electrical Engineering, College of Engineering, Prince Sattam bin Abdulaziz University, Al-Kharj, 11942, Saudi Arabia
[c] ENSA, Sidi Mohammed Ben Abdellah University, Fez, 30000, Morocco
[d] Department of Electrical Engineering, Faculty of Engineering, Al-Baha University, Alaqiq, KSA, Saudi Arabia
[e] Wolfson Centre for Magnetics, School of Engineering, Cardiff University, Cardiff, CF24 3AA, UK
[f] Electrical Engineering Department, Faculty of Engineering, Al-Baha University, Al-Baha, 65779, Saudi Arabia

ARTICLE INFO

Keywords:
Wind energy conversion system
FPGA
DFIG
Proportional integral derivative controller
HDL
FPGA-In-the-loop
MATLAB/Simulink

ABSTRACT

This study employs an FPGA board to implement a robust control technique for wind energy conversion systems (WECS). This approach facilitates extensive testing and validation of the control system across diverse wind conditions, utilizing the FPGA's parallel processing capabilities and advanced control algorithms. This method ensures robustness against nonlinearities and system uncertainties. FPGA-in-the-loop (FIL) testing provides precise and effective simulation results, enabling rapid prototyping and iterative modifications of control algorithms. The effectiveness of the robust control strategy is confirmed by FIL findings, demonstrating significant improvements in WECS stability and efficiency. Furthermore, the study highlights the strategy's potential to enhance WECS reliability and efficiency in real-world applications.

1. Introduction

In the past, energy demand was low, resulting in limited interest in utilizing renewable energy sources. However, in today's world, electricity significantly affects both domestic and commercial activities. Renewable energy systems are in great demand because they improve energy security, lower greenhouse gas emissions, and support sustainable power sources. Wind energy is unique among renewable energy sources because of its enormous potential and low environmental impact, along with solar, hydro, biomass, and other sources.

The Doubly-Fed Induction Generator (DFIG) is a highly favored generator within the Wind Energy Conversion Systems (WECS) family due to its exceptional energy conversion efficiency and adaptable operation. DFIGs are great for capturing wind energy because they maintain constant rotor speeds, even when the wind fluctuates. Also, DFIG systems are cost-effective compared to other systems and perform well even when wind speed changes [1,2]. To ensure efficient energy transfer and keep the grid stable, it's crucial to control active and reactive power robustly [3,4].

Many control strategies have been developed and applied to wind energy systems to meet the challenges posed by the non-linear and time-varying nature of wind energy systems. We describe a limited number of examples in this section, such as Fuzzy Logic

* Corresponding author.
E-mail address: badre.bossoufi@usmba.ac.ma (B. Bossoufi).

https://doi.org/10.1016/j.heliyon.2024.e35712
Received 9 February 2024; Received in revised form 16 July 2024; Accepted 1 August 2024
Available online 3 August 2024

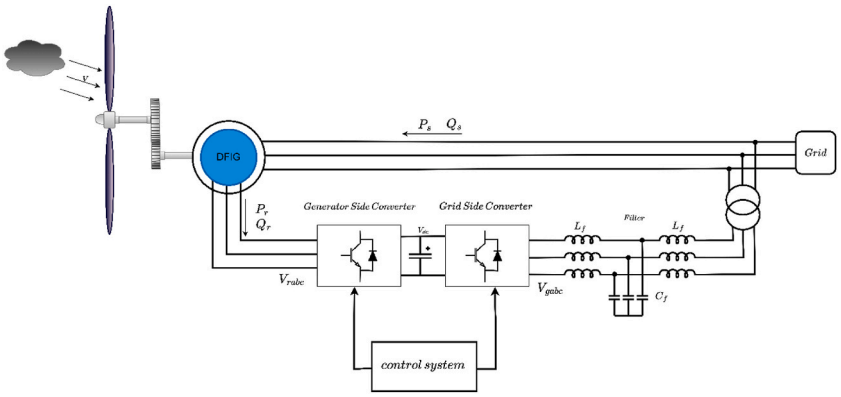

Fig. 1. Architecture of the system.

Control, Model Predictive Control (MPC), Sliding Mode Control (SMC), Adaptive Control and Field-Oriented Control (FOC).

Fuzzy Logic Control, this is a nonlinear control strategy that makes use of a set of membership functions to handle the system uncertainties and nonlinearity. The major merit of fuzzy logic control is its capability of handling system nonlinearities and uncertainties without requiring a precise mathematical model of the system. While this may garner several benefits, the design and tuning process of fuzzy logic controllers are complicated, and performance depends heavily on the quality of fuzzy rules. Moreover, fuzzy logic controllers may have a problem in computational inefficiency, and it doesn't guarantee the optimal control performance each time [5].

Model Predictive Control (MPC) predicts the future behavior of a system by using a dynamic model that optimizes control actions within a finite horizon to achieve the required performance. Besides multivariable systems handling and constraints, MPC is very adaptable to providing optimal control actions based on future predictions. However, it has some disadvantages: it is a computer-intensive technique that requires an accurate system model. Design complexity of the MPC algorithm itself, together with its real-time computational demands, significantly increase the level of difficulty over its application to systems with fast dynamics [6].

Sliding mode control (SMC) is a robust control technique that forces, through discontinuous control actions, the system state to slide along a predefined surface. It ensures a high level of robustness with parameter variations and disturbances and is relatively easy to implement. However, one major drawback of this technique is the chattering phenomenon, which may be responsible for wear in mechanical components; therefore, it usually requires smoothing techniques. Moreover, designing an appropriate sliding surface is very tricky, and this approach is also inadequate to perform well under all operating conditions [7].

Adaptive control is a way to adjust the parameters in real-time for counteracting changes in system dynamics and external disturbances, so that systems with time-varying parameters have good performance and adaptability to varying operating conditions. However, adaptive control normally involves a complicated design procedure together with a stability analysis, and usually has a slower response in comparison with the fixed-parameter controllers. Besides these, the need for continuous parameter adjustment may increase the computational burdens still further [8].

Field-Oriented Control, also known as vector control, is another advanced control technique primarily being applied to control AC motors. In this control technique, FOC-independent controls are developed, which decouple the control of torque and flux. This provides very accurate and dynamic control of the motor's performance and hence finds its application in those applications where a high level of accuracy in control and efficiency is required, viz. WECS. The main advantage of FOC is that it provides better transient and steady state performance along with improved efficiency and handling of nonlinearities and parameter variations. However, implementing the FOC technique involves complex transformations and requires an accurate knowledge of motor parameters, hence it is difficult to implement and requires sophisticated hardware and software [9].

Sudden disturbances and unexpected changes in wind conditions can affect the lifecycle and efficiency of WECS. Robust control is crucial to keep the system operational in situations of uncertainty and external interference. To ensure the stability and reliability of WECS across various operational conditions, this approach is essential [10]. Robust control not only ensures stable and efficient energy production, but also enhances fault tolerance and optimizes energy extraction.

Recently, FPGAs have been utilized to enhance the efficiency of wind energy systems. These programmable integrated circuits execute control algorithms quickly and with great flexibility. FPGAs are reconfigurable, making them suitable for dynamic applications, and thus, FPGA is a suitable technology for Wind Energy Conversion. The use of FPGAs in control systems for wind energy conversion serves to enhance stability and performance. Several reviews have emphasized the advantages of FPGA implementations in various applications due to their high computational capabilities [11–13].

The use of FPGA-in-the-loop (FIL) testing has proven to be highly effective in evaluating control strategies developed for wind energy systems. During the development process, control techniques can be tested and validated by integrating FPGA-in-the-loop. FIL ensures the stability of control mechanisms while also accelerating the design and prototyping process. Previous research [14,15], and [16] has emphasized the essential role of FPGA-in-the-loop testing in verifying the performance and efficiency of renewable energy applications.

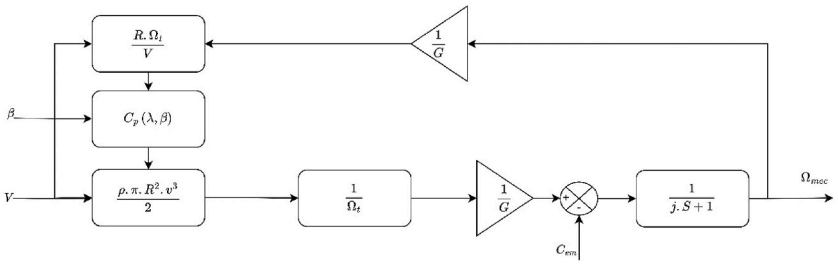

Fig. 2. Model of a wind turbine.

This paper is organized as follows: Section 2 describes the model for the wind turbine and explains the key structures and processes involved in its operation. Section 3 discusses the FPGA board, focusing on its architecture, functions, and relevant operations. Section 4 discusses the FPGA-in-the-loop testing of the system, including testing procedures, results, and analysis. Section 5 presents conclusions on the paper's theme and explores the potential expansion of wind energy conversion systems through FPGA technologies in the future.

2. Model of the wind turbine

Fig. 1 shows the structure of the system studied.

2.1. Modeling of the aeromechanical part

The equation of wind power is given by eq. (1) [17]:

$$Pv = \frac{\rho.s.v^3}{2} \tag{1}$$

where:
ρ: Density of the area (1225 kg/m3).
v: wind speed.
s: the turbine surface (π. R^2).
Only part of the wind's energy can be converted by a wind turbine [17,18]. This is denoted by C_p (λ, β) as a function of *the* λ speed ratio and β blade angle.
The turbine's aerodynamic power [17,18] is derived from eq. (2):

$$P_{aero} = \frac{C_p(\lambda,\beta)\cdot\pi\cdot R^2\cdot\rho\cdot v^3}{2} \tag{2}$$

The power coefficient C_p is defined as the wind turbine's aerodynamic efficiency [18]. Depending on the turbine's characteristics [17,18] it can be defined by eq. (3) and eq. (4):

$$C_p(\lambda,\beta) = C_1\left(\frac{C_2}{A} - C3\cdot\beta - C_4\right) * \exp\left(-\frac{C_5}{A}\right) + C_6 * \lambda \tag{3}$$

With:

$$\frac{1}{A} = \frac{1}{(\lambda + 0.08*\beta)} - \frac{0.035}{\beta^3 + 1} \tag{4}$$

$C_1 = 0.5179$; $C_2 = 116$; $C_3 = 0.4$; $C_4 = 5$; $C_5 = 21$; and $C_6 = 0.0068$.

Equation (5) calculates the speed ratio, defined as the ratio of the turbine's linear speed Ω_t to the wind speed v [17,18]:

$$\lambda = R\cdot\frac{\Omega_t}{v} \tag{5}$$

The aerodynamic torque is given by eq. (6):

$$C_{aero\ 0} = \frac{C_p(\lambda,\beta)\cdot\rho\cdot\pi\cdot R^2\cdot v^3}{2\cdot\Omega_t} \tag{6}$$

The figure above presents the total wind turbine model (Fig. 2).

2.2. Multiplier model and mechanical shaft model

The gearbox is responsible for converting the slow-speed turbine to a fast-speed generator. It can be represented mathematically by the following equations [19]:

$$\begin{cases} \Omega_t = \dfrac{\Omega_{mec}}{G} \\ Cg = \dfrac{C_{aero\ 0}}{G} \end{cases} \tag{7}$$

2.3. Model

2.3.1. Three-phase electrical space representation of a DFIG

The three-phase electrical space representation of DFIGs provides a comprehensive framework for analyzing and understanding the electrical behaviour of these systems. By representing the stator and rotor windings as vectors in a three-dimensional space, this representation enables the visualization and interpretation of key electrical quantities such as voltages, currents, and fluxes. Furthermore, it facilitates the development and optimization of control strategies aimed at enhancing the performance, stability, and efficiency of DFIG-based wind energy conversion systems.

The machine voltages, currents and flux can be described by the following equations.

➢ Equation (8) for the stator and 9 for the rotor voltages are expressed as follows:

$$(V_s) = [R_S](I_S) + \frac{d}{dt}(\Phi_S) \tag{8}$$

$$(V_R) = [R_R] \cdot (I_R) + \frac{d}{dt}(\Phi_R) \tag{9}$$

where R_S, R_R are the stator and rotor resistances, respectively.

I_S, I_R currents of the stator and rotor.

➢ *The Flux expression is shown in* eq. (10 _11)*:*

$$(\Phi_S) = [L_S] \cdot (I_S) + [M_{SR}] \cdot (I_R) \tag{10}$$

$$\begin{gathered} (\Phi_R) = [L_R] \cdot (I_R) + [M_{SR}]^T \cdot (I_S) \\ [L_S] = \begin{bmatrix} l_S & m_S & m_S \\ m_S & l_S & m_S \\ m_S & m_S & l_S \end{bmatrix} \\ [M_{SR}] = m_{SR} \cdot \begin{bmatrix} \cos\alpha & \cos(\alpha - 4\Pi/3) & \cos(\alpha - 2\Pi/3) \\ \cos(\alpha - 2\Pi/3) & \cos\alpha & \cos(\alpha - 4\Pi/3) \\ \cos(\alpha - 4\Pi/3) & \cos(\alpha - 2\Pi/3) & \cos\alpha \end{bmatrix} \end{gathered} \tag{11}$$

2.3.2. PARK representation of DFIG

Park transformation is a mathematical method for transforming the 3-phase output of an asynchronous machine into a two-coordinate reference frame. This makes it easier to analyze and control machines in rotating reference frames [20].

For each of the vectors defined above (voltage and flux), we will change the PARK coordinate system on both the stator and rotor (eq. (12)).

$$[P(\theta_S)] = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos\theta_S & -\sin\theta_S & \dfrac{1}{\sqrt{2}} \\ \cos(\theta_S - 2\Pi/3) & -\sin(\theta_S - 2\Pi/3) & \dfrac{1}{\sqrt{2}} \\ \cos(\theta_S - 4\Pi/3) & -\sin(\theta_S - 4\Pi/3) & \dfrac{1}{\sqrt{2}} \end{bmatrix} \tag{12}$$

where $[P(\theta_s)]$ *and* $[P(\theta_s)]^{-1}$ represent the direct and inverse passing matrices of PARK, respectively.

➢ *The equations for the stator voltages* V_{SP} eq. (13) *and the rotor* V_{RP} are:

$$(V_{SP}) = [R_S] \cdot (I_{SP}) + \frac{d}{dt}(\Phi_{SP}) + [P_1(\theta_S)]^{-1} \cdot \frac{d}{dt}([P_1(\theta_S)]) \cdot (\Phi_{SP}) \tag{13}$$

This is easily demonstrated by multiplying the matrices in eq. (14).

$$(V_{sP}) = [R_S](I_{SP}) + \frac{d}{dt}(\Phi_{SP}) + \omega_S \cdot \begin{bmatrix} -\Phi_{Sd} \\ +\Phi_{Sq} \\ 0 \end{bmatrix} \tag{14}$$

Or by developing PARK components for the stator eq. (15):

$$\begin{cases} V_{Sd} = R_S \cdot I_{Sd} + \dfrac{d}{dt}\Phi_{Sd} - \omega_S.\Phi_{Sq} \\ V_{Sq} = R_S \cdot I_{Sq} + \dfrac{d}{dt}\Phi_{Sq} + \omega_S \cdot \Phi_{Sd} \end{cases} \tag{15}$$

where:

V_{Sd}, V_{Sq}: d/q stator voltages.

I_{Sd}, I_{Sq}: d/q stator currents.

We follow the same steps to calculate the PARK components for the rotor (eq. (16)):

$$\begin{cases} V_{Rd} = R_R \cdot I_{Rd} + \dfrac{d}{dt}\Phi_{Rd} - \omega_R.\Phi_{Rq} \\ V_{Rq} = R_R \cdot I_{Rq} + \dfrac{d}{dt}\Phi_{Rq} + \omega_R \cdot \Phi_{Rd} \end{cases} \tag{16}$$

where:

V_{Sd}, V_{Sq}: d/q rotor voltages.

I_{Sd}, I_{Sq}: d/q rotor currents.

➢ *Flux expression*

For the calculation of Flux, similar steps are used, as shown below in eq. (17).

$$(\Phi_{SP}) = [\mathrm{P}(\theta_S)]^{-1} \cdot [L_S]\Big[[\mathrm{P}(\theta_S)](I_{SP}) + [\mathrm{P}(\theta_S)]^{-1} \cdot [M_{SR}]\Big][\mathrm{P}(\theta_R)](I_{RP}) \tag{17}$$

By expanding each line, we obtain the PARK components of the stator eq. (18) flux and rotor flux in equation (19) and equation (20), respectively:

$$\begin{cases} \Phi_{Sd} = L_S \cdot I_{Sd} + M_{SR} \cdot I_{Rd} \\ \Phi_{Sq} = L_S \cdot I_{Sq} + M_{SR} \cdot I_{Rq} \end{cases} \tag{18}$$

$$\begin{cases} \Phi_{Rd} = L_R \cdot I_{Rd} + M_{SR} \cdot I_{Sd} \\ \Phi_{Rq} = L_R \cdot I_{Rq} + M_{SR} \cdot I_{Sq} \end{cases} \tag{19}$$

➢ *The expression of electromagnetic torque*

Equation (20) defines the electromagnetic torque:

$$\mathrm{T_{em}} = \mathrm{p}.\left(\Phi_{\mathrm{sd}}.\,\mathrm{I_{sq}} - \Phi_{\mathrm{sq}}.\,\mathrm{I_{sd}}\right) \tag{20}$$

3. Indirect field-oriented control

It is clear from looking at the aforementioned expressions that a flux term and a current term interact to produce electromagnetic torque [21]. Notably, there is a strong similarity between these expressions and the torque expression in a direct current machine. However, the challenge lies in controlling the flux term and the current term separately. This degree of independence is necessary to maximize the performance of the direct current machine and, consequently, to achieve accurate control over its output torque [22,23]. Several strategies can be employed to achieve this control. Here, we describe stator flux-oriented control, choosing to align the stator's flux with the d-axis as shown in Eq. (21), [18,24].

$$\begin{cases} \Phi_{\mathrm{sd}} = \Phi_{\mathrm{s}} \\ \Phi_{\mathrm{sq}} = 0 \end{cases} \tag{21}$$

Given that R_S is insignificant, as the losses from joule heating are small compared to the losses for the medium and high-power machines utilized in wind energy conversion, we obtain eq. (21):

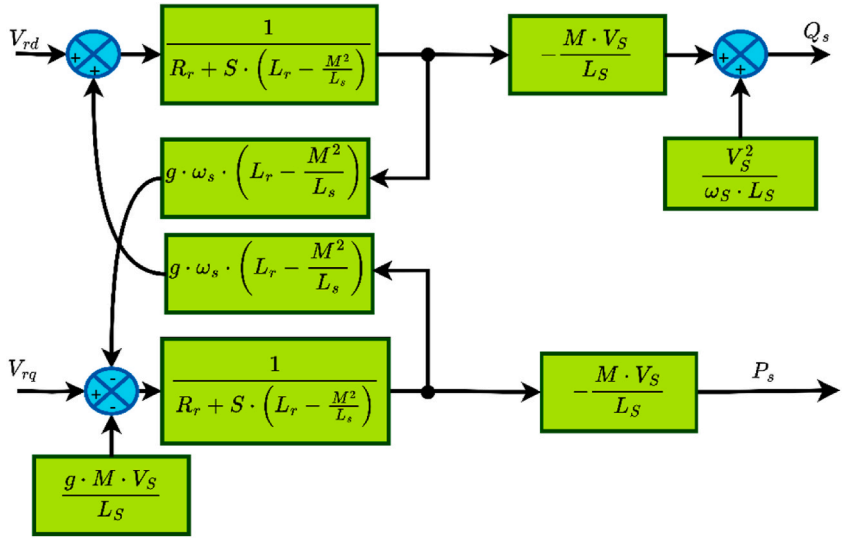

Fig. 3. The block diagram of the DFIG for the electrical system.

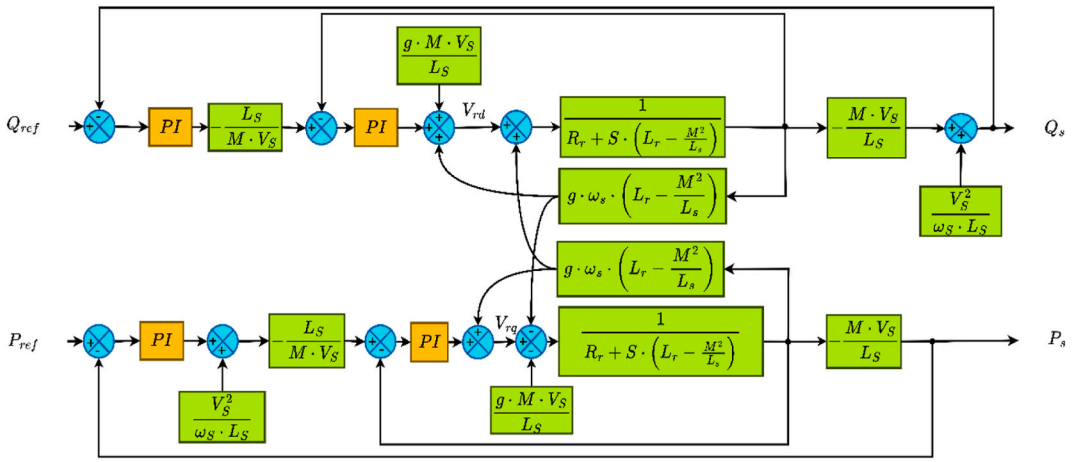

Fig. 4. Block of the controlling diagram for the DFIG.

$$\begin{cases} V_{Sd} = \dfrac{d\Phi_S}{dt} \\ V_{Sq} = \omega_S \cdot \Phi_{Sd} \end{cases} \tag{21a}$$

In this case, stator voltages become as follow eq. (22):

$$\begin{cases} \mathrm{V_{sd}} = 0 \\ \mathrm{V_{sq}} = \mathrm{V_s} = \omega_s.\Phi_s \end{cases} \tag{22}$$

From equation (12), eq. (23) is obtained:

$$\begin{cases} I_{Sd} = \dfrac{1}{L_S}(\Phi_S - M_{SR} \cdot I_{Rd}) \\ I_{Sq} = \dfrac{M_{SR}}{L_S} \cdot I_{Rq} \end{cases} \tag{23}$$

By integrating equation (23) into equation (16), the expression of rotor voltage becomes (eq. (24))

$$\begin{cases} V_{rd} = \left(R_r + S.\left(L_r - \dfrac{M_{SR}{}^2}{L_s}\right)\right).I_{rd} - \omega_s.g.\left(L_r - \dfrac{M_{SR}{}^2}{L_s}\right).I_{rq} \\ V_{rq} = \left(R_r + S.\left(L_r - \dfrac{M_{SR}{}^2}{L_s}\right)\right).I_{rq} + \omega_s.g.\left(L_r - \dfrac{M_{SR}{}^2}{L_s}\right).I_{rd} + \dfrac{g.M_{SR}.V_s}{L_s} \end{cases} \tag{24}$$

Fig. 3 depicts the block diagram of the electrical system derived from the two equations above (eq. (24)). First-order transfer functions connect the rotor and stator voltages and powers, as shown in the block diagram. This configuration allows for independent control of each axis, facilitating a vector control system. Each axis has its own controller, operating independently without cross-coupling. Consequently, the active power for the rotor axis q and the reactive power for the rotor axis d serve as the reference values for these controllers.

To achieve power control of this machine, two types of control structures were employed. Direct flux orientation control (DFOC) is the first approach; it simplifies by ignoring coupling terms and uses independent controllers for each axis to control reactive and active power separately [25]. The second approach uses a two-loop system to manage both power and rotor currents, and it incorporates a robust control model that takes coupling factors into account. By using this method, the rotor currents driving the DFIG are precisely controlled.

In this paper, we are interested in a robust control method; our choice is based on existing studies that compare the two structures. However, the advantages of this method include the possibility of double correction and the ability to couple P and Q together. Fig. 4 shows the controlling diagram for the DFIG (Fig. 4).

The open-loop transfer function H(S) can be represented by the following equation (eq. (25)):

$$H(S) = \left(\frac{1}{R_r + S.\left(L_r - \frac{M_{SR}{}^2}{L_s}\right)}\right).\left(K_{P1} + \frac{K_{I1}}{S}\right) \tag{25}$$

The compensation method removes 0 from the transfer function *H(p)*. To ensure that the *H (p)* function maintains the stable performance of the compensating process, the following transfer function parameters are presented in eq. (26):

$$\frac{K_{P1}}{K_{I1}} = \frac{L_r - \frac{M_{SR}{}^2}{L_s}}{R_r} \tag{26}$$

From equation (27), the open loop function of transfer *H(S)* is presented as follows:

$$H(S) = \frac{K_{I1}}{S.R_r} \tag{27}$$

The closed-loop function of transfer *G(S)* can be defined by eq. (28):

$$G(S) = \frac{1}{1 + S.\frac{R_r}{K_{I1}}} \tag{28}$$

which is equivalent to this structure in eq. (29):

$$G(S) = \frac{1}{1 + S.\tau} \tag{29}$$

where τ is the system response time.

The current corrector parameters, expressed as a function of machine parameters and response time, are as follows:

$$K_{I1} = \frac{R_r}{\tau} \quad ; \quad K_{P1} = \frac{L_r - \frac{M_{SR}{}^2}{L_s}}{\tau} \tag{30}$$

For calculating the PI's controller parameter, we used the same method as for the function of machine parameters and response time. In the end, we obtain the result shown by eq. (31):

$$K_{I0} = \frac{1}{\tau_0} \quad ; \quad K_{P0} = \frac{\tau_0}{\tau} \tag{31}$$

where τ_0 is the system response time for the reactive power control loop.

4. FPGA

A Field-Programmable Gate Array (FPGA) is a semiconductor device that can solve a wide range of computing problems. They are embedded circuits that can be customized by the customer or designer after manufacturing [26]. FPGAs consist of a matrix of Customizable Logic Blocks (CLBs) connected by programmable interconnects. These versatile components are essential for various

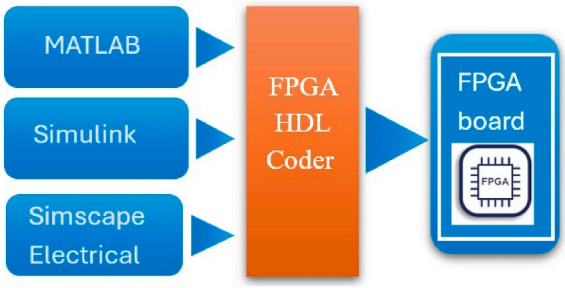

Fig. 5. FIL illustration.

applications, including computing, telecommunications, and industry [27].

FPGA architectures, such as matrix, memory-based, anti-fuse-based, hybrid, and programmable logic gates, provide different design options. Matrix architecture offers high flexibility with interconnected logic blocks. Memory-based FPGAs rely on SRAM for speed but require continuous power. Anti-fuse-based FPGAs enhance security, while hybrid architectures combine different technologies. Programmable logic gate architectures are suitable for low-power or small-scale designs [28].

FPGAs consist of several key components that work together to provide flexibility and customization. The main components of an FPGA include the following [29].

- A look-up table or LUT is used to perform logic operations.
- Off-chip memory controllers
- The Flip-Flop is a register element where the LUT results are stored.
- I/O pads, allowing the entry and output of data to the FPGA.
- High-speed serial transceivers
- Embedded memories for distributed data storage
- DSP_48 blocks
- PLLs or phase-locked loops for driving the FPGA fabric at different clock rates.
- Wires, connecting the elements.

The components mentioned above, combined with reprogramming, make FPGAs powerful tools for implementing custom electronic circuits, rapid prototyping, and developing embedded systems. Designers are offered unprecedented flexibility to adapt their designs to the specific needs of their projects.

FPGA evolution is moving towards embedding microprocessors within their circuits. This shift in architecture may result in the development of reconfigurable systolic architectures. Coarse-grained reconfigurable architectures, which already exist, show promise as technology advances. High-end FPGAs with 40 nm SRAM technology can contain billions of transistors and perform thousands of multiplications within nanoseconds.

4.1. FPGA-in-the-loop

"FPGA-in-the-Loop (FIL) is a method primarily utilized in the realms of digital design and embedded systems for testing and validating software algorithms and hardware designs. This approach involves integrating field-programmable gate arrays (FPGAs) into system simulation and testing processes to leverage the high performance and flexibility offered by FPGAs."

In an FPGA, "digital representation formats" are used for both data processing and data transfer. Some basic formats for representations in computing include floating-point, fixed-point, and two-complement formats. The floating-point format is used to represent real numbers. While fixed-point number processors can handle multiple values and perform numerous calculations, they require extensive hardware support. Modern FPGAs offer embedded floating-point services for complex mathematical operations, but they also require substantial hardware resources. The integer format is typically chosen using two's complement representation because both positive and negative integer formats appear the same within this binary representation. This format is suitable for representing both positive and negative values. Additionally, the fixed-point format is commonly used in implementing FPGAs and other DSP modules to encode numbers in digital systems. In the fixed-point format, the decimal point has a fixed location, but the number of digits that can appear after the decimal point may vary. However, fixed-point representations are more beneficial in regions that frequently require precise measurement. These formats consist of specific numbers of bits for the integer and fractional parts, making them suitable for accurate calculations. In our study, we utilized fixed-point representation for certain hardware on the discretized FPGA to implement optimization.

Similarly, electronic test interfaces play a crucial role in validating, programming, and troubleshooting FPGAs. The Joint Test Action Group (JTAG) is one of the most popular electronic test interfaces for FPGAs, designed as an industry-standard method for testing ICs within large and complex systems. It operates on a chain-based design model, sequentially connecting each circuit component, such as the FPGA. This JTAG chain ensures that the necessary test pins for testing, debugging, and other operations are

Table 1
Parameters of the DFIG generator and WT.

DFIG			WT		
Parameters	Symbols	Values	Parameters	Symbols	Values
Power generator (kW)	P_S	1.5 M	**Radius of the turbine blade (m)**	R	35.25
Pole number	P	2	**Turbine and generator moment (N.m)**	J	10
Stator resistance (Ω)	R_S	1.2	**Density of air (kg/m^3)**	ρ	1.225
Rotor resistance (Ω)	R_r	1.8	**Tip-speed ratio**	λ_{opt}	8
Stator inductance (mH)	L_S	0.1554	**Optimal power coefficient**	C_P	0.4942
Rotor inductance (mH)	L_r	0.1568	-	–	–

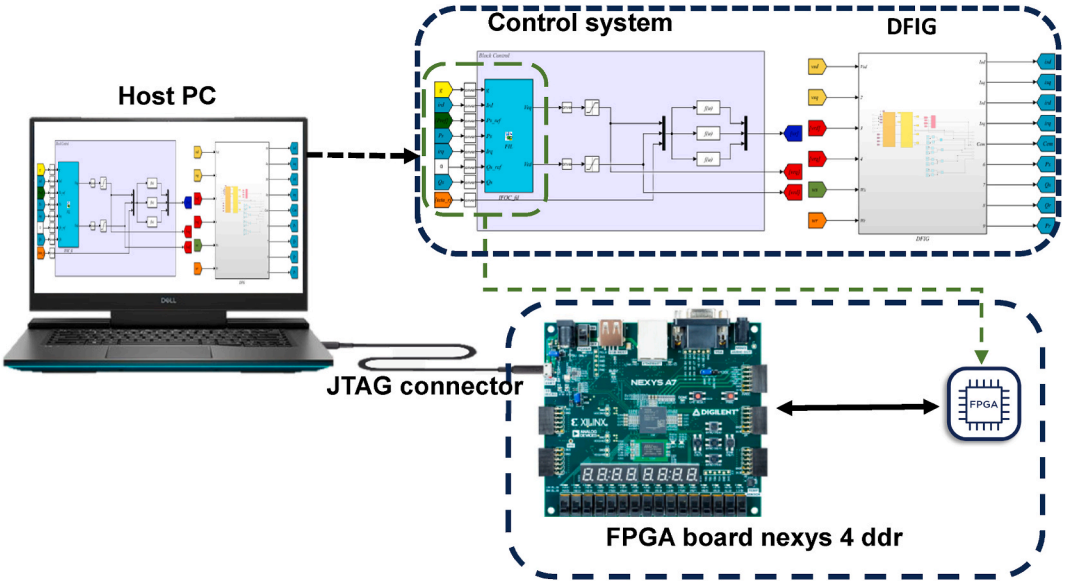

Fig. 6. FPGA in the loop test of proposed method using FPGA board nexys 4 ddr.

easily accessible. As discussed earlier, JTAG offers a significant advantage over other interfaces in terms of diagnostics, allowing probing or remote access to FPGAs. This feature is particularly useful for managing FPGAs embedded in remote locations, enabling remote programming and debugging. It also facilitates access to FPGA configuration information, aiding in the reading of programming and detecting potential programming errors. In our simulation, we utilized JTAG test interfaces for identification. Depending on the application's requirements, other connection interfaces such as USB or Ethernet may replace or complement JTAG.USB interfaces are prevalent in most systems, providing a straightforward and convenient method for connecting FPGAs to computers and other devices. Widely used for programming FPGAs and data transfer between FPGAs and host processors, memory is characterized by low access time and high transfer rates. USB is an easy-to-use solution that works right out of the box and does not require deep configuration or setting adjustments from the end user. It is mainly used in applications under development or for simple prototyping. Thus, it is suitable for situations where large volumes of data need to be transmitted at high rates over long distances. Full-duplex mode is utilized so that data can be both received and sent within the FPGA. This capability is crucial for adaptive control systems, sensor networks, and other time-sensitive applications (see Fig. 5).

In general, FPGA-in-the-loop (FIL) refers to a broad development method integrating both FPGA and software in the loop techniques, using tools like Vivado and Simulink (see Fig. 5). This method proposes a global strategy for creating and designing complicated structures of digital systems rather than updating them when needed. It begins with creating a system model in Simulink, where the system under investigation is represented as functional and data flow blocks organized hierarchically [30]. The model is then tested with various scenarios and inputs during simulation to replicate real-world conditions and relationships. Engineers can utilize Simulink's simulation engine to detect errors, observe design behavior, and even make minor adjustments before progressing to the final stage of hardware implementation.

5. Results and discussion

Table 1 presents the DFIG and wind turbine parameters employed in our simulation. However, to create an FIL setup in Simulink, several steps are involved in establishing the connection between the FPGA hardware and MATLAB. Fig. 6 presents the flow diagram of our study. Below are the key steps to set up the FPGA in the loop using Simulink.

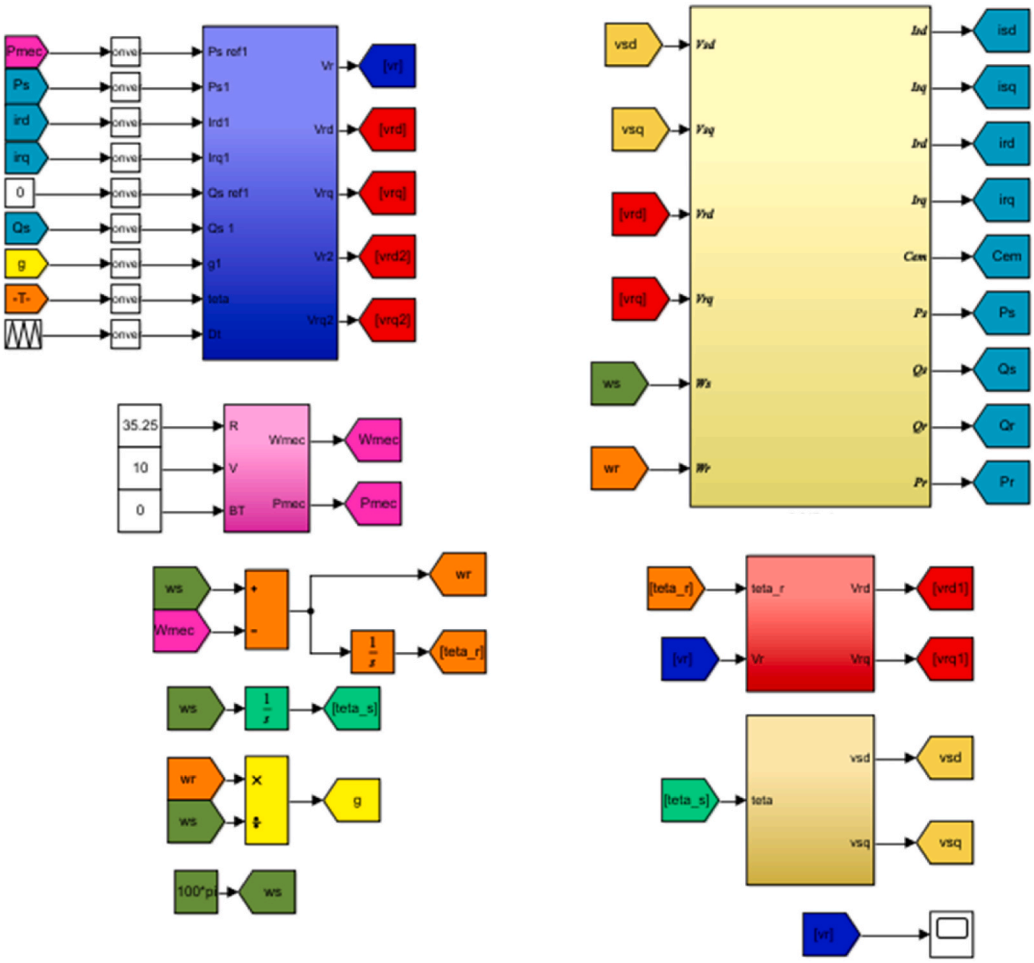

Fig. 7. Model representation in Simulink.

- The FPGA design software is configured on the system before using FPGA-in-the-loop. To add the Vivado design software to the system path for the current MATLAB session, we used the **hdlsetuptoolpath** function.
- The FPGA in the Loop Wizard is launched by performing the following steps: Select the FIL Wizard from the Code Verification, Validation, and Test area in the Apps gallery.
- Specify if the wizard will generate a FIL Simulink block or a FIL Simulation MATLAB System Object. For this example, **Simulink** for the **FIL simulation** is selected.
- The HDL design that will be used in the FPGA was defined.
- JTAG was selected as the connectivity method.

In Simulink, the system model shown in Fig. 7 is generated. The central yellow block represents the DFIG generator, the key element in power generation. The system is controlled via the blue block to ensure stability and control. The wind turbine model, represented by the pink block, simulates the operation of the wind energy source. The other blocks are dedicated to important calculations that add to the model's complexity, such as the calculation of angular velocity (w) and park transformation (vsd and vsq), to determine the system's stator and rotor components. This Simulink system provides interesting information on the interactions between these components for studying and optimizing the performance of DFIG generators and wind turbines, demonstrating its potential in the research and development of advanced energy systems. In addition, we integrated pipeline optimization into our FPGA implementation for the wind turbine control system to enhance the overall efficiency and performance. This enhancement allows for more streamlined processing and improved responsiveness in managing wind turbine operations.

Fig. 8 below shows the robust control system introduced in the indirect field-oriented control section. This implementation is a key transition between theory and practical application, enabling the testing and implementation of the approach.

Fig. 9 represents the Simulink block of the total system of the FPGA in the loop in our study, within its inputs and outputs considered.

The Simulink model is first converted into hardware-specific code using the HDL Coder tool after successful simulation. This

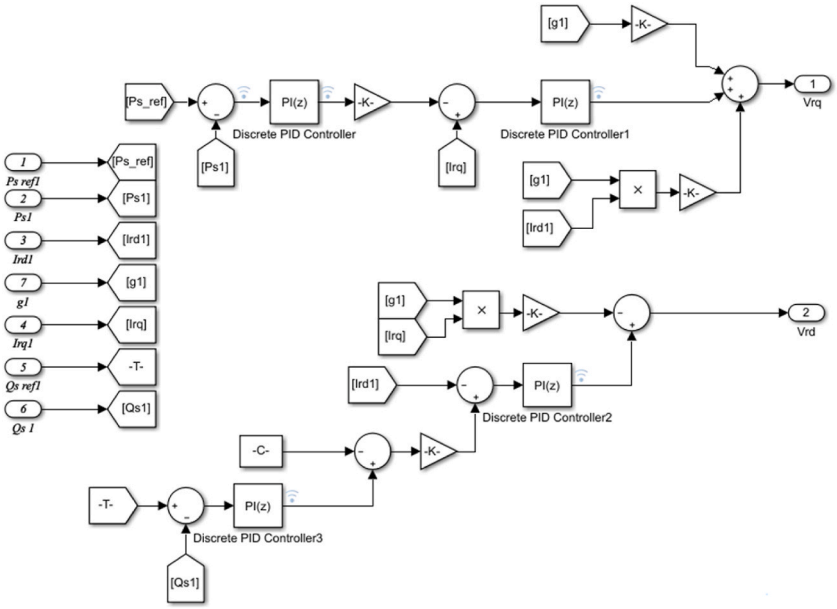

Fig. 8. Representation of the robust control system in Simulink.

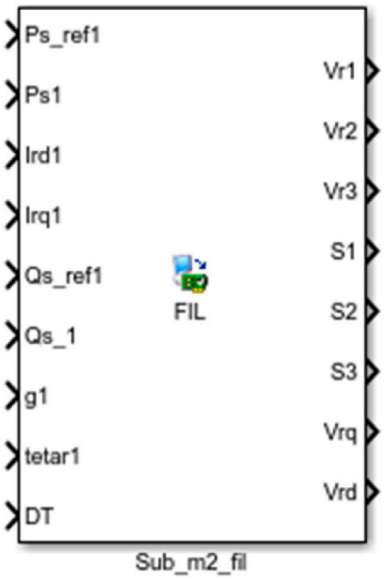

Fig. 9. Block of the global system of the FPGA in the Simulink loop.

generated HDL code represents the system's behavior in a hardware description language like VHDL or Verilog. Then comes Vivado, which processes the HDL code through steps such as synthesis and place-and-route to change the abstract design into a configuration bitstream that can be loaded onto the target FPGA hardware. This is where the true power of FPGA-in-the-loop comes into play. Fig. 11 provides an overview of the digital schematic of this approach. By converting the model into HDL, we create a direct hardware representation, allowing the system to be deployed and run on an FPGA, offering dedicated hardware acceleration for model calculations. The main advantage here is the improved performance and parallelization of operations, inherent features of FPGAs, making this approach especially appealing for applications that require real-time calculations and increased performance.

When programming MATLAB/Simulink models on an FPGA with the "FPGA-in-the-Loop" test, different types of peripherals are added to ensure communication and control between the FPGA and MATLAB/Simulink. These peripherals include serial interfaces such as Ethernet interfaces for communication via the network and FPGA-internal standards such as AXI to simplify communication between processing blocks. JTAG peripherals are often used for debugging, while GPIO pins offer versatile digital connectivity. This

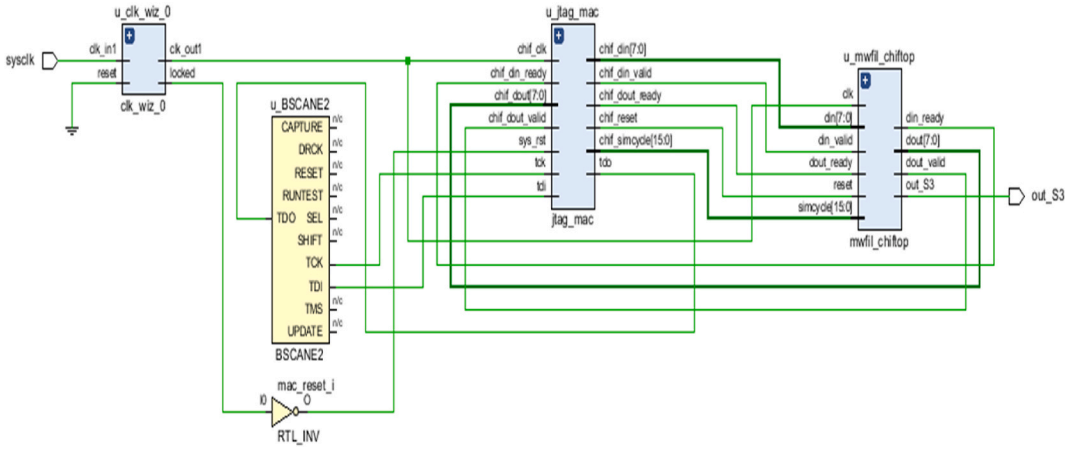

Fig. 10. The global numeric model in Vivado responsiveness.

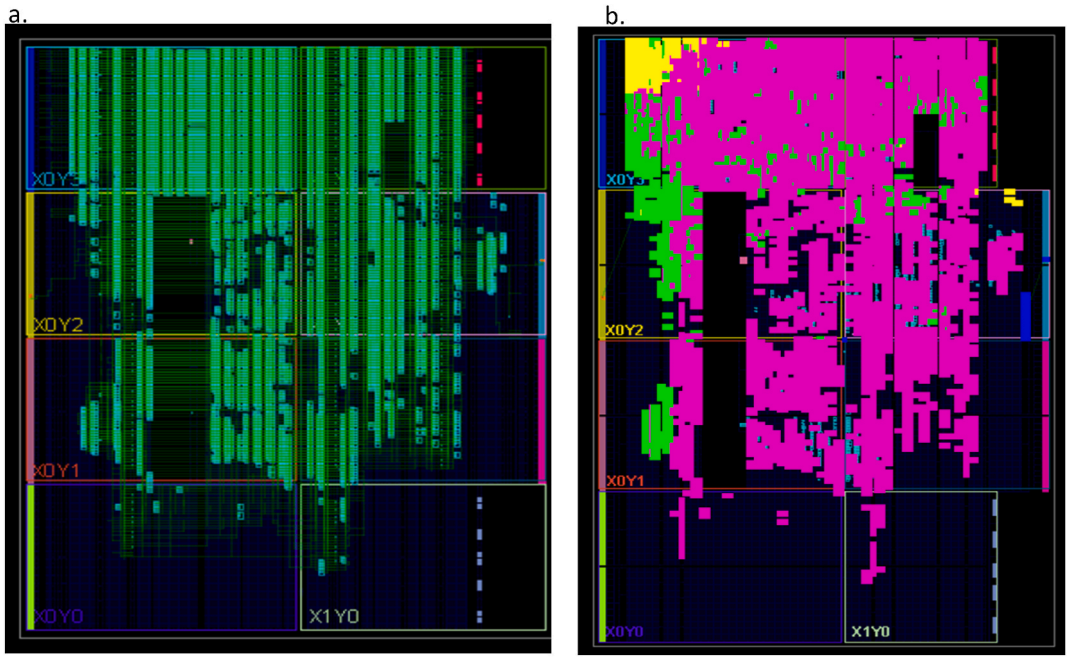

Fig. 11. FPGA architecture. (a) FPGA routing. (b) FPGA placement.

diversity of peripherals allows them to interact with the FPGA control model and the software development environment to debug and collect data.

In this research, the JTAG periphery is utilized, which enables bidirectional communication with electronic components through a specific set of pins. This allows for non-intrusive modification of the internal state of circuits. JTAG is commonly applied in production testing, debugging electronic circuits, and programming programmable devices. It serves as a crucial tool for maintaining and validating complex electronic equipment. Fig. 11 illustrates the FPGA architecture of the control system. In Fig. 11b, the pink area represents the HDL system from Fig. 10, with the remaining parts representing the communication interfaces.

The active power and its reference are depicted in Fig. 12, showing close similarities. This indicates that the responses of the active and reactive power match the reference power from the wind turbine. The resemblance demonstrates that the control system effectively follows the reference power. The alignment between the actual and desired outputs serves as evidence of the accurate and reliable performance of the control mechanism, ensuring that the wind turbine operates within specified parameters. This alignment serves as an indicator of the control system's effectiveness in power regulation, ultimately contributing to the stability and efficiency of the wind turbine.

The similarity between the active/reactive power and reference power curves in this figure is clear proof that the wind turbine power and FPGA power are identical. This matching of two outputs improves the accuracy and precision of the control system and

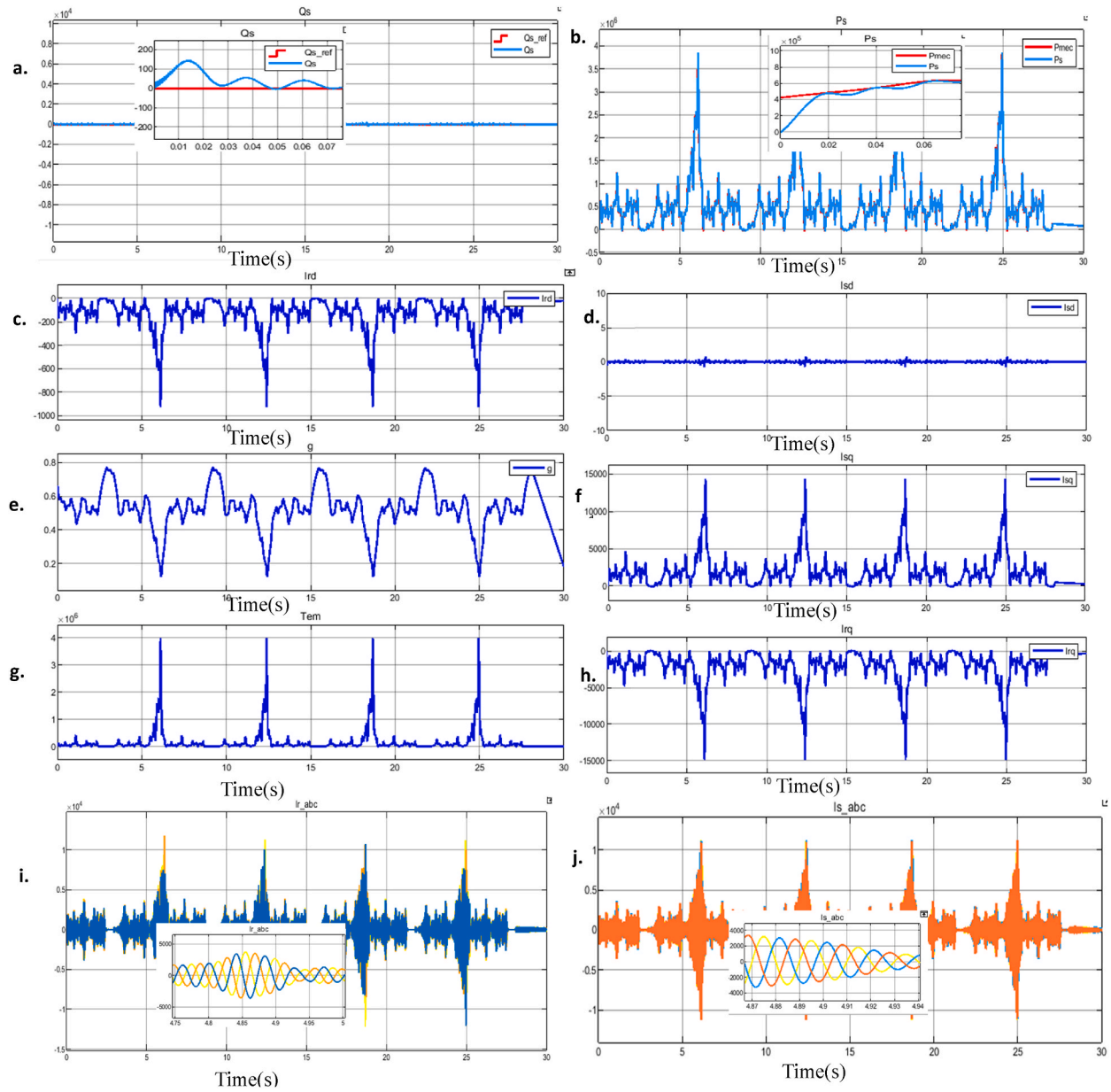

Fig. 12. Simulation results with variable wind speed. (a) Stator reactive power. (b) Stator active power. (c) rotor direct current Ird. (d) stator direct current Isd. (e) Slip g. (f) stator quadrature current Isq. (g) electromagnetic torque Tem. (h) rotor quadrature current Irq. (i) Rotor current ir_abc. (j) Stator current is_abc.

ensures that the wind turbine output is exactly as per the specifications. Finally, the match between the actual and desired outputs proves not only the efficiency of the control system but also that the FPGA is replicating the wind turbine energy production. This level of similarity guarantees accurate and reliable performance of our FPGA model.

The stator and rotor voltage signals are represented in Fig. 12. Vrd varies as a function of the stator's reactive power Qs, and Vrq varies within stator's active power Ps. Fig. 12a and 12b illustrate this relationship. Therefore, it can be concluded that the generator tracks both active and reactive powers accurately.

In the Park coordinate system, the direct axis is chosen to coincide with the stator magnetic field. The reference axis, therefore, follows the stator flux and is chosen to minimize the component of the alternating current passing through it. Most loads and electrical systems are designed to operate most efficiently when the direct axis current is maintained at or near zero, which facilitates system regulation and control.

The vsd component (direct stator voltage) is set equal to zero, as the goal is to minimize magnetic flux variations in this axis. Any voltage or current in this axis could disrupt system stability. On the other hand, the vsq component (stator quadrature voltage) results

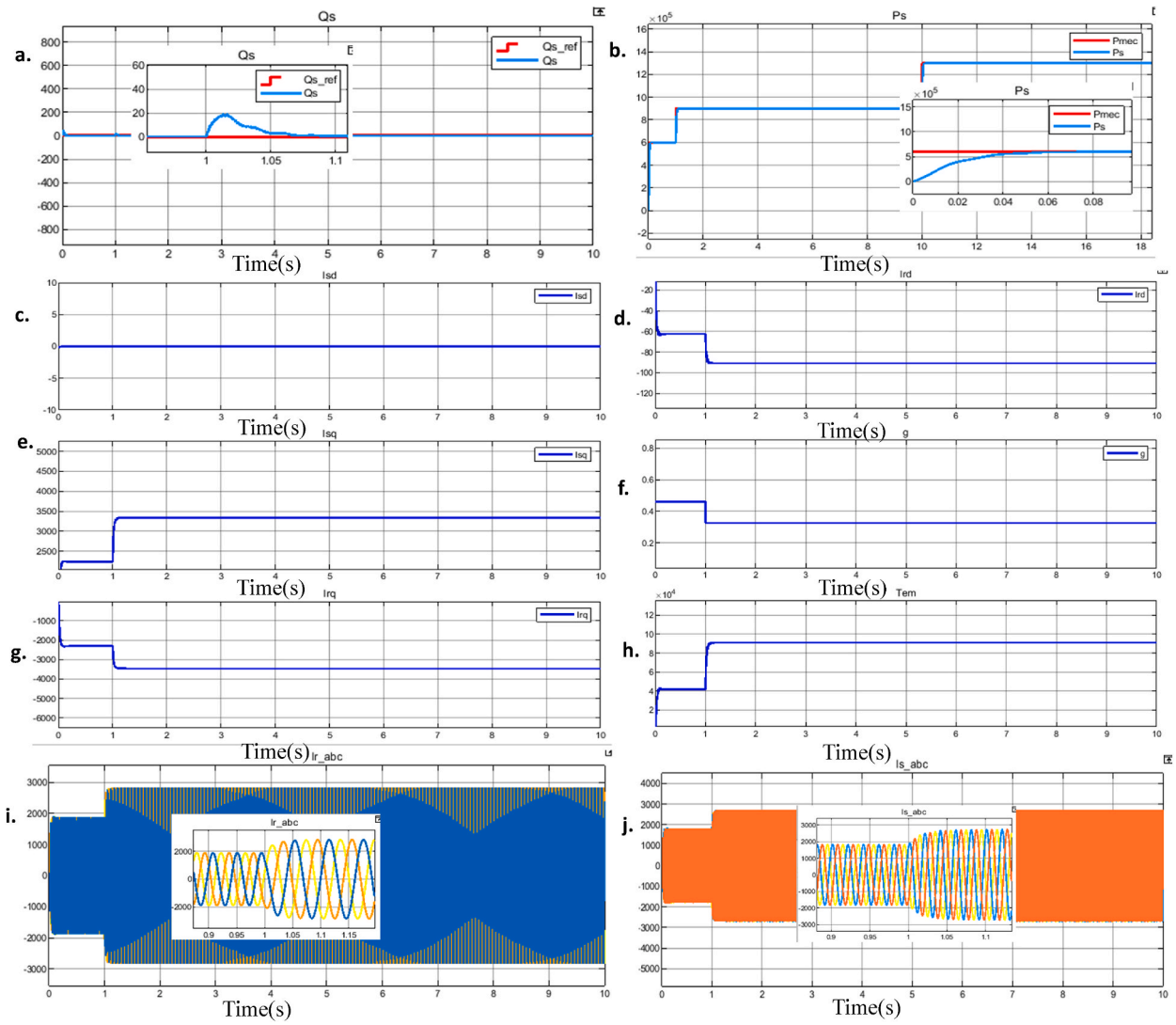

Fig. 13. Results of the simulation with constant wind speed. (a) Stator reactive power. (b) Rotor active power. (c) stator direct current Isd. (d) rotor direct current Ird. (e) stator quadrature current Isq. (f) Slip g. (g) rotor quadrature current Irq. (h) electromagnetic torque Tem. (i) Rotor current ir_abc. (j) Stator current is_abc.

from the projection of the voltage on the direct axis onto the perpendicular axis, generally referred to as the quadrature axis. This projection is necessary to control the machine's magnetic flux and torque, which is essential for correct operation of the electric machine.

Fig. 13c, Fig. 13h, Fig. 13d and Fig. 13 f present the direct rotor current Ird, the quadrature rotor current Irq, the direct stator current Isd and the quadrature stator current Isq respectively. Control strategies incorporate Ird and Irq to manage machine performance, power generation, and system stability. The interaction between Ird, Irq, Vrd, and Vrq is crucial for precise control, ensuring efficient and reliable operation across all applications.

Fig. 13 displays simulation results under constant parameters. Comparing these with results under variable parameters (Fig. 12) demonstrates the robustness, adaptability, and flexibility of the control strategy. In Fig. 14, with constant parameters, the system remains stable, maintaining consistent performance under unchanged conditions. Conversely, Fig. 12 shows results under variable wind speeds ranging from 3.5 to 14.5 m/s, offering a dynamic perspective on the system's response to environmental and operational changes. This comparison highlights the importance of parameter adaptability in ensuring effective and resilient wind turbine control systems, particularly in fluctuating real-world conditions. Robustness tests involved increasing Rr and Rs by 100 % of their nominal values, with results detailed in Fig. 14.

The simulation results depicted in Fig. 14 demonstrate the resilience of our control strategy when subjected to parametric variations of the DFIG. This is proven by the consistent adherence to power set points, accompanied by uniform response times during start-up. Moreover, the control mechanism effectively maintains the decoupling of active and reactive powers throughout the

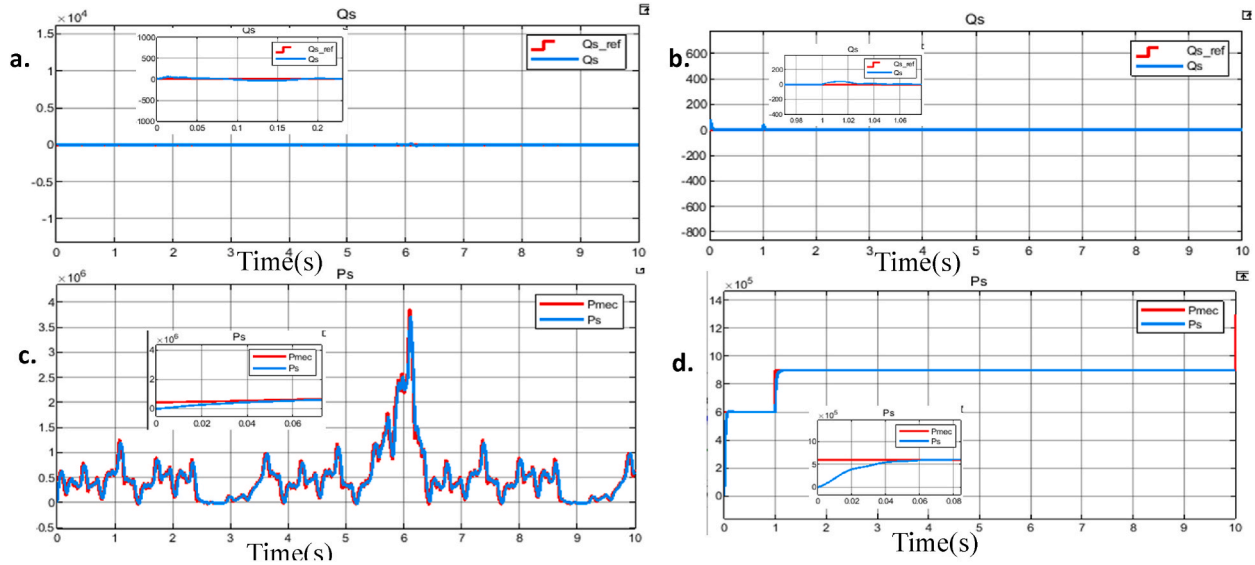

Fig. 14. Variation of the stator and rotor resistance. (a) Stator active power at a constant wind speed. (b) Rotor reactive power at a constant wind speed. (c) Stator active power at a variable wind speed. (b) Rotor reactive power at a variable wind speed.

Table 2
Performance comparison.

	Used method	Efficiency	% Error	% Overshot	Robustness
[31]	Backstepping adaptative	98.99	0.12	0	high
[32]	Adaptive fuzzy vector controller	93.5	0.15	0	moderate– high
[33]	Direct torque control (DTC)-classical	92.13	0.32	5	moderate– high
[34]	Sliding mode	94.82	0.2	0	low
Proposal Technique		99.01	0.1	0	high

operation.

Based on the results of our simulation, the proposed technique has provided excellent results compared to other control techniques [31–34] (Table 2), in terms of efficiency, error and robustness.

6. Conclusion and perspectives

In this article, a comprehensive study was conducted involving the simulation of a wind turbine model in the MATLAB/Simulink environment to adjust power output. The methodology involved the construction of a Simulink model, the generation of HDL code from it, and its seamless integration into an FPGA board to enhance system responsiveness. The results demonstrate a high degree of agreement between Simulink and FPGA-in-the-loop (FIL) tests, validating the robustness and applicability of the approach across simulations.

Quantitatively, simulation time was significantly reduced with FPGA implementation compared to traditional Simulink simulations. Minimal discrepancy in output power adjustments between FPGA and Simulink was observed, underscoring the accuracy and reliability of the FPGA approach. Furthermore, improved responsiveness and computational efficiency were noted. In summary, the effectiveness and feasibility of FIL testing for wind turbine control applications are highlighted, with potential avenues for further research and industrial applications in renewable energy systems.

Future research aims to integrate artificial neural networks (ANNs) into the wind turbine control model, replacing conventional PID controllers to enhance system performance and adaptability. This evolution towards AI-based control seeks to leverage ANNs' learning capabilities for advanced wind turbine operation control. A comparison of PID and ANN control techniques will be conducted to investigate their performance under varying conditions, showcasing AI's potential in optimizing renewable energy systems.

CRediT authorship contribution statement

Abdelhafid El Attafi: Writing – original draft, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Houda El Alami:** Resources, Formal analysis, Data curation. **Badre Bossoufi:** Visualization, Validation, Supervision. **Dokhyl AlQahtani:** Software, Resources, Project administration, Methodology, Investigation. **Saad Motahhir:** Visualization, Validation, Supervision. **Mishari Metab Almalki:** Validation, Software, Resources. **Thamer A.H. Alghamdi:** Project administration,

Methodology, Investigation, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] S. Afifi, H. GholamHosseini, R. Sinha, FPGA implementations of SVM classifiers: a review, SN Computer Science (2020), https://doi.org/10.1007/s42979-020-00128-9.
[2] J. Gs, M. P. M, D.M.Z. Kurian, FPGA-Based Implementations– A Review, IJIREEICE, 2022, https://doi.org/10.17148/ijireeice.2022.10231.
[3] H. Yang, J. Zhang, J. Sun, L. Yu, Review of advanced FPGA architectures and technologies, J. Electron. (2014), https://doi.org/10.1007/s11767-014-4090-x.
[4] K. Vipin, S.A. Fahmy, FPGA dynamic and partial reconfiguration, ACM Comput. Surv. (2019), https://doi.org/10.1145/3193827.
[5] D. Patel, V. Kumar, High-performance control of wind energy systems using FPGA technology, Journal of Sustainable Energy Systems 9 (4) (2018) 355–368, https://doi.org/10.1109/JSES.2018.8614102.
[6] A.R. Gokhale, R.B. Patil, FPGA implementation of power electronic converter for DFIG based wind turbine, Energy Proc. 90 (2016) 206–214, https://doi.org/10.1016/j.egypro.2016.11.200.
[7] R. Williams, S. Brown, Evaluation of robust control algorithms using FPGA-in-the-loop testing, J. Control Eng. Technol. 30 (2) (2023) 210–225, https://doi.org/10.1016/j.jcet.2022.10.005.
[8] H. He, Q. Shi, Real-time simulation and hardware-in-the-loop testing: a survey of trends and techniques, IEEE Trans. Ind. Inf. 15 (9) (2019) 5054–5065, https://doi.org/10.1109/TII.2019.2904039.
[9] S. Bolognani, M. Zigliotto, FOC and DTC: two viable schemes for induction motors torque control, IEEE Trans. Power Electron. 17 (5) (2002) 779–787, https://doi.org/10.1109/TPEL.2002.801925.
[10] C. Nagamani, G.S. Ilango, M.A.A. Rani, A. Prasanthini, Computation of Rotor Position of DFIM Using Rotor Side Phase Locked Loop, 2017, https://doi.org/10.1109/NPSC.2016.7858861.
[11] P.M. Menghal, A. Jaya Laxmi, Real Time Control of Electrical Machine Drives: A Review, 2010, https://doi.org/10.1109/ICPCES.2010.5698697.
[12] D. Raveendhra, B. Kumar, D. Mishra, M. Mankotia, Design of FPGA based open circuit voltage MPPT charge controller for solar PV system, Proc. IEEE Int. Conf. Circuit, Power Comput. Technol. ICCPCT 2013 (2013), https://doi.org/10.1109/ICCPCT.2013.6529012.
[13] K. Ni, C. Gan, Y. Hu, D.T. Lagos, R. Qu, N.D. Hatziargyriou, Emulated stator voltage-oriented vector control of DFIM-SPS with coupling effect elimination for electric ship applications, IEEE Trans. Transp. Electrif. (2020), https://doi.org/10.1109/TTE.2020.3039355.
[14] M. Debbou, M. Pietrzak-David, P. Cussac, Torque Ripple Minimization of SMC Drive for an Innovative Electric Naval Propulsion System, 2015, https://doi.org/10.1109/EPE.2015.7311712.
[15] Y. Mokhtari, D. Rekioua, High performance of maximum power point tracking using ant colony algorithm in wind turbine, Renew. Energy (2018), https://doi.org/10.1016/j.renene.2018.03.049.
[16] M.W. Naouar, E. Monmasson, A.A. Naassani, I. Slama-Belkhodja, N. Patin, FPGA-based current controllers for AC machine drives - a review, IEEE Trans. Ind. Electron. (2007), https://doi.org/10.1109/TIE.2007.898302.
[17] A. Tarfaya, D. Dib, M. Ouada, S. Ghoudelbourk, Simple and effective control and optimization of a wind turbine based on a DFIG, Int. J. Comput. Aided Eng. Technol. (2021), https://doi.org/10.1504/IJCAET.2021.115349.
[18] B. Bossoufi, M. Karim, A. Lagrioui, M. Taoussi, A. Derouich, Observer backstepping control of DFIG-Generators for wind turbines variable-speed: FPGA-based implementation, Renew. Energy (2015), https://doi.org/10.1016/j.renene.2015.04.013.
[19] S. Ganjefar, A. Mohammadi, Variable speed wind turbines with maximum power extraction using singular perturbation theory, Energy (2016), https://doi.org/10.1016/j.energy.2016.03.095.
[20] A. El Fadili, F. Giri, A. El Magri, L. Dugard, Nonlinear Controller for Doubly Fed Induction Motor with Bidirectional AC/DC/AC Converter, 2013, https://doi.org/10.3182/20130703-3-FR-4038.00040.
[21] M. Aktas, K. Awaili, M. Ehsani, A. Arisoy, Direct torque control versus indirect field-oriented control of induction motors for electric vehicle applications, Eng. Sci. Technol. an Int. J. (2020), https://doi.org/10.1016/j.jestch.2020.04.002.
[22] I. Saady, et al., Optimization for a photovoltaic pumping system using indirect field oriented control of induction motor, Electron (2021), https://doi.org/10.3390/electronics10243076.
[23] A. Iqbal, B.P. Reddy, S. Rahman, M. Meraj, Modeling and indirect field-oriented control for pole phase modulation induction motor drives, IET Power Electron. (2023), https://doi.org/10.1049/pel2.12381.
[24] N. Cherfia, D. Kerdoun, Wind energy conversion systems based on a DFIG controlled by indirect vector using PWM and SVM, Int. J. Electr. Comput. Eng. (2016), https://doi.org/10.11591/ijece.v6i2.7858.
[25] D.W. Novotny, Implementation of direct stator flux orientation control on a versatile dsp based system, IEEE Trans. Ind. Appl. (1991), https://doi.org/10.1109/28.85484.
[26] A. Proulx, J.Y. Chouinard, P. Fortier, A. Miled, A Survey on FPGA Cybersecurity Design Strategies, ACM Trans. Reconfigurable Technol. Syst., 2023, https://doi.org/10.1145/3561515.
[27] Y.S. Dandass, S.C. Burgess, M. Lawrence, S.M. Bridges, Accelerating string set matching in FPGA hardware for bioinformatics research, BMC Bioinf. (2008), https://doi.org/10.1186/1471-2105-9-197.
[28] X. Yue, C. McLeod, FPGA design and implementation for EIT data acquisition, Physiol. Meas. (2008), https://doi.org/10.1088/0967-3334/29/10/007.
[29] T. Ohkawa, K. Yamashina, H. Kimura, K. Ootsu, T. Yokota, FPGA components for integrating FPGAs into robot systems, IEICE Trans. Inf. Syst. (2018), https://doi.org/10.1587/transinf.2017RCP0011.
[30] Y. Huo, G. Gruosso, Hardware-in-the-Loop framework for validation of ancillary service in microgrids: feasibility, problems and improvement, IEEE Access (2019), https://doi.org/10.1109/ACCESS.2019.2914346.
[31] B. Bossoufi, et al., DSPACE-based implementation for observer backstepping power control of DFIG wind turbine, IET Electr. Power Appl. (2020), https://doi.org/10.1049/iet-epa.2020.0364.
[32] Z. Zhang, Y. Zhao, W. Qiao, et al., A space-vector-modulated sensorless direct-torque control for direct-drive PMSG wind turbines, IEEE Trans. Ind. Appl. 50 (4) (2014) 2331–2341.
[33] A. Zemmit, S. Messalti, A. Harrag, A new improved DTC of doubly fed induction machine using GA-based PI controller, Ain Shams Eng. J. (2016) 1–9.
[34] S. Abderazak, N. Farid, Comparative study between sliding mode controller and fuzzy sliding mode controller in a speed control for doubly fed induction motor, IEEE 4th Int. Conf. in Control Engineering & Information Technology, Hammamet, Tunisia (December 2016) 1–6.