IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 8 May 2023, accepted 29 May 2023, date of publication 6 June 2023, date of current version 15 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3283420

RESEARCH ARTICLE

An Experimental Hybrid Control Approach for Wind Turbine Emulator

MUHAMMAD AKRAM BHAYO[1], SOHRAB MIRSAEIDI[2], (Senior Member, IEEE), MOHSIN ALI KOONDHAR[1], SAADULLAH CHANDIO[1], MOHSIN ALI TUNIO[3], HAITER LENIN ALLASI[4], (Member, IEEE), MOHD JUNAIDI ABDUL AZIZ[5], (Member, IEEE), AND NIK RUMZI NIK IDRIS[5], (Senior Member, IEEE)

[1]Department of Electrical Engineering, Quaid-e-Awam University of Engineering, Science and Technology, Nawabshah 67480, Pakistan
[2]School of Electrical Engineering, Beijing Jiaotong University, Beijing 100044, China
[3]Department of Electrical Engineering, Mehran University of Engineering and Technology SZAB Campus, Khairpur Mir's 66020, Pakistan
[4]Department of Mechanical Engineering, Kombolcha Institute of Technology, Wollo University, Kombolcha 208, Ethiopia
[5]School of Electrical Engineering, Faculty of Engineering, Universiti Teknologi Malaysia, Johor Bahru 81310, Malaysia

Corresponding author: Sohrab Mirsaeidi (msohrab@bjtu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 52150410399.

ABSTRACT In this study, the experimental implementation of a Hybrid Controller (HC) for a Wind Turbine Emulator (WTE) is presented. Conventionally, the speed control of a WTE is based on two Proportional Integral (PI) controllers used for the outer (speed) loop and inner (torque) loop of a DC motor drive. However, based on the extensive literature survey, with the PI controllers alone, significant overshoot and substantial settling time in the speed response are unavoidable. The overshoot cannot be completely removed without sacrificing speediness. To eliminate the overshoot and concurrently improve the settling time, an HC, combination of Adaptive Neuro-Fuzzy Inference System (ANFIS) and PI controller for the outer and inner loops, respectively, is proposed, designed, implemented and analyzed. The actual wind turbine is emulated using a 2-quadrant DC motor drive based on DS1104 R&D controller board from dSPACE. These two control schemes, namely PI (both controllers are PI), and HC (PI controller for inner loop and ANFIS controller for outer loop) for the WTE are extensively simulated based on the small-scale wind turbine model and eventually implemented using a small-scale laboratory set-up. When implemented in the WTE, with a step-change in wind speed, the hybrid control scheme produces a better result. It is found out the Root Mean Square Error (RMSE) for the WTE rotor speed of the HC is 0.495, which is lower compared to PI control scheme RMSE of 0.7273. Finally, to verify the validity of the WTE based on HC, a random wind speed with wind turbine parameters extracted from the actual wind turbine is applied. The results indicated that the WTE perfectly emulated the wind turbine rotor speed with an RMSE of 0.5004.

INDEX TERMS Wind turbine emulator (WTE), ANFIS controller, PI controller, DS 1104 R&D Controller Board, wind energy conversion system (WECS), hybrid controller.

I. INTRODUCTION

In today's modern era, Wind Energy Conversion System (WECS) has become one of the developing technologies used globally. It is a promising future technology to suffice the electrical energy demand. Compared to other renewable energy sources, wind energy has expanded rapidly [1]. The primary source of any WECS is the wind, it is intermittent, it varies haphazardly, and it is uncontrollable. Consequently, a wind turbine provides variable mechanical output power. Therefore, the electrical generator attached to a wind turbine's shaft also produces variable electrical power [2]. Hence, it is imperative to analyze a real WECS before the installation at the site. Subsequently, to investigate the behavior of a WECS before its installation, the development of a WTE becomes unavoidable. Most significantly, in real WECS, it would be nearly impossible to directly test or validate the control algorithms, because of the above

The associate editor coordinating the review of this manuscript and approving it for publication was Jason Gu.

mentioned issues and the cost. The WTE, a laboratory-based test-bench that replicates the behavior of real WECS, also helps researchers to investigate the performance of the power electronic converters of the real WECS.

The development of a WTE would also be helpful in designing and validating the control strategies and algorithms. Hence WTE offers the engineers, researchers, and designers a platform for conducting research, observing and analyzing the operation and control of WECS, in a controlled environment in the absence of natural wind and without an actual wind turbine [1]. A WTE can also be developed aiming especially to provide a platform for the generator control algorithm [3]. Additionally, a WTE can be utilized by the students as a teaching tool [4]. It can be useful for the purpose to limit the torque and power of variable wind speed, pitch control, and generator control [5].

In designing a WTE, the actual turbine rotor is replaced by an electric motor. The motor reproduces the characteristics of an actual wind turbine. The motor is driven by a controlled power converter. Based on the literature survey, a motor (ac or dc), a PC with the necessary software installed, an intelligent power module, a controller, and a feedback mechanism are the main components of any WTE [6], [7], [8], [9]. Whereas, remaining parts, i.e. the generator (dc generator, PMSG, or DFIG), power electronics circuits, and the electrical load may remain common in both the real system as well as emulation system. Therefore, in a WTE, a motor drive physically implements the static and dynamic properties of a wind turbine.

The existing literature provides information on the properties of wind turbines and their mathematical modeling [6], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19]. Different models of aerodynamic analysis of wind turbines can be found in [20]. A detailed literature review has shown that several researchers have utilized different motor types, such as separately excited dc motors, Permanent Magnet Synchronous Motor (PMSM), squirrel cage induction motors, and servomotors, etc. Furthermore, it has been noted that in the past, researchers often developed WTE based on a dc motor [6], [9], [10], [11], [12], [13], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40] with current control (i.e. torque control on its shaft). Although the dc motors are bulky in size, they are comparatively expensive but the control system is relatively easy. Also, from the perspective of control, the dc motor offers an easy control approach, in which the torque and speed can easily be controlled through armature current and voltage from zero to maximum speed [41]. In reference [42] permanent magnet dc motor driven by thyristor controlled rectifier has been used and the authors have built the prototype with Pulse Width Modulation (PWM) controlled dc-dc converter. In references [43], [44] the real wind turbine is replaced by PMSM. The authors of [45] have also chosen PMSM for simulating the wind turbines. In reference [3] the emulator is physically implemented by using two servomotors connected back-to-back. In reference [46] a PMSM directly coupled to the Induction Generator (IG) is deployed as the prime mover to emulate the wind turbines operation. As reported in reference [11] the researchers of [47] and [48] used Insulated Gate Bipolar Transistor (IGBT) based inverter controlled Induction Motor (IM). In [49], [50], [51], [52], [53], [54], [55], [56], [57], and [58] as well IM is used. Even though, IM is smaller in size as compared to dc motors, they are less expensive when compared with the dc motors. However, the control system of IM is comparatively complex. Apart from complexity, IM is not a good choice for emulation of low wind speed and power. The authors in reference [59] have used a squirrel cage induction machine coupled with a permanent magnet induction generator. Literature survey shows that, many classes of WTEs may be distinguished based on how they were constructed, such as dc-machine-based WTEs and those based on the Hardware-In-the-Loop (HIL) concept [60], [61]. To simulate the dynamic behavior of an actual wind turbine, it mostly consists of closed loops connecting physical and software components.

There are different controllers reported in the literature, including PID controllers. From the literature survey, it is found that for WTE controllers, majority of the researchers [7], [8], [11], [39], [51], [55], [62], [63], [64], [65], [66] have used PI controllers. However, the use of PI controllers in WTE is inadequate. This is because PI's performance is poor. The controller's limited steady gain range is the main cause of this mediocre performance. In addition, a PI controller's small gain causes a slow response. Particularly in WTE, it takes a lot of time to set the PI controllers' parameters, which have non-linearity (because of random wind speed). Therefore, using a PI controller is not a good choice for WTEs. Consequently, an alternative solution is compulsory. Though PI (or PID) controllers are used in WTEs due to zero steady-state error and good stability. But still, there are some issues, like, optimum tuning of the K_p, K_i, and K_d and a significant amount of overshoot and settling time. PID controllers' tuning, however, is a challenging issue, particularly when the plants are unstable and nonlinear. Although the various PI controller tuning strategies can enhance the response of PI controllers, it becomes more difficult and time-consuming.

The 2DofPID and fractional PID are available in the literature. Therefore, to overcome the limitations of conventional PID controllers, more advanced control techniques such as 2DofPID and fractional PID have been developed and reported in the literature. Furthermore, 2DofPID is a modified version of the conventional PID controller that introduces a feed forward control term and a low-pass filter to improve the control performance. Whereas, fractional PID extends the conventional PID controller by introducing a fractional-order term that allows for better control of processes with long time delays and nonlinearity. However, both 2DofPID and fractional PID have their drawbacks and limitations in wind turbine emulator system. One major limitation of 2DofPID is

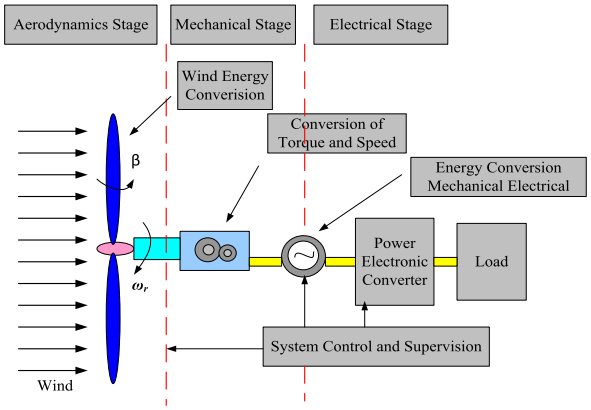

FIGURE 1. The wind energy conversion system.

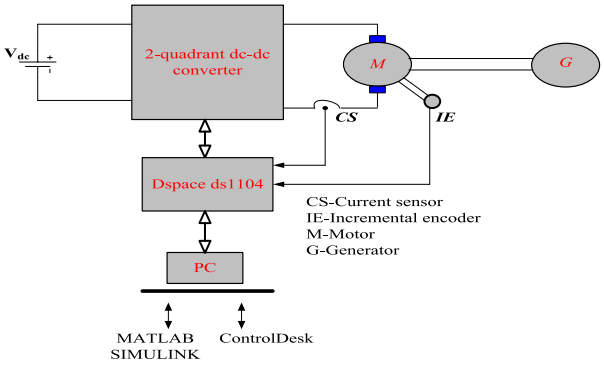

FIGURE 2. The proposed wind turbine emulator.

its increased complexity and the need for accurate modeling and parameter tuning, which may not be practical in WTE applications. Fractional PID is also computationally intensive and requires accurate modeling and parameter tuning, which is not appropriate for WTE.

Generally, for tuning of the PID controllers, not specifically for WTE controllers, a variety of optimization techniques can be used, e.g., Particle Swarm Optimization (PSO), modified PSO, Genetic Algorithm (GA), Ant Bee Colony (ABC), Salp Swarm Algorithm (SSA) etc. Furthermore, the computational intelligence techniques e.g., Artificial Immune System (AIS), Artificial Neural Networks (ANN), Fuzzy Logic Control (FLC) technique, Adaptive Neuro-Fuzzy Inference System (ANFIS), may also be used. These techniques can solve those complex problems, which cannot be solved by the conventional methods with desired speed and accuracy. Therefore, the authors of [67] and [68] used FLC for the control of the motor. Nonetheless, for WTE controllers, if the rules for FLC are defined incorrectly, the WTE cannot accurately replicate the behavior of real WECS. It should also be noted that FLC is not an expert by itself. It relies upon the rule values defined by the user. In reference [69] ANN is proposed.

Therefore, in this research study, the performance analysis of PI controller based WTE is studied both in simulation and hardware. Despite PI controllers' advantages, these controllers have encountered the problem of parameter tuning. Therefore, for WTE controllers, an alternative solution is compulsory. Hence, PI controller is replaced with an Adaptive Neuro-Fuzzy Inference System (ANFIS) based control system. Moreover, varying wind profile is the primary factors that should be considered when constructing an accurate WTE. Therefore, in this research study, a random wind speed signal based on a real wind speed profile reported by [21] is generated using MATLAB.

Furthermore, arbitrary changes in wind speed, such as gusts of wind, abrupt increases or decreases in wind speed, and gradual changes in wind speed should also be taken into consideration during modeling, so that the designed controller can be tested. Therefore, in this research study, these factors have been considered and the wind speed test signal is generated in MATLAB. Because of the inaccurate and slow response of the PI controllers, they must be improved by using an alternate option. The existing wind turbine emulators, based on PI controllers, may not accurately simulate the power output of a real wind turbine. This can lead to inaccurate results when testing and evaluating wind turbines, which can negatively impact the design and development of wind turbines. Hence, to address the above-mentioned issues, in this research study, hybrid control approach is proposed to improve the performance of the controller for WTE. A hybrid controller is proposed, designed, simulated, and implemented experimentally using the DS1104 R&D Controller Board.

II. PROPOSED WIND TURBINE EMULATOR

The WECS considered in this research study for developing a WTE is shown in Figure 1. In terms of modeling, the WECS can be divided into three stages, namely, the aerodynamic, mechanical, and electrical stages. It is imperative to highlight that amongst all the wind turbines commercially installed worldwide, approximately 70% of them comprise 3-bladed Horizontal Axis Wind Turbines (HAWT).

Hence, in this research study, the HAWT based WECS has been selected for modeling and analysis purposes. Figure 2 shows the proposed WTE.

The wind turbine is emulated using a dc motor, which is fed by a 2-quadrant dc-dc converter. The speed and the torque of the wind turbine are emulated by controlling the speed and torque of the dc motor (labeled as M in the figure) using a closed-loop cascaded control structure. The reference speed for the dc motor (to emulate the turbine speed) is generated based on the real-time implementation of the wind turbine model using the DS1104 controller board from dSPACE. The wind turbine model requires the motor speed and armature current (i.e. torque); they are obtained using an incremental encoder (labeled as IE in the figure) and a hall-effect current sensor (labeled as CS in the figure), respectively. In this setup, G represents the dc generator. For designing the 2-Q dc-dc power converter, an IGBT module is used. It is selected because the dc motor used in this research study is required to run only in one direction. Therefore, a 2-Q dc-dc converter is appropriate. The DS1104 R&D Controller Board is used to implement these two control methods.

A. dSPACE DS1104 R&D CONTROLLER BOARD

The dSPACE DS1104 R&D Controller Board is the core component of the developed WTE. The controller board used in this research study is a comprehensive real-time control system, which is based on a 603 PowerPC floating-point processor running at 250 MHz. It is a versatile commercial system with input-output interfaces and a real-time processor on one board which is inserted directly into a PC. For extended input-output purposes, the controller board comprises a slave-DSP subsystem based on the TMS320F240 DSP microcontroller. This DS1104 R&D Controller Board is sufficient enough to perform multiple tasks, e.g. computing a reference signal (WTE model), generation of PWM signals, analog to digital conversion, and as well as for acquiring the rotational speed. For the applications of rapid control prototyping (RCP), specific interface connectors and specially designed connector panels facilitate simple access to all input and output signals of the board. The use of the DS1104 R&D Controller Board has upgraded the PC to a prevailing development system for RCP. Thus, the DS1104 R&D Controller Board is the ideal hardware for the design and realization of this proposed WTE. This board contains eight 16-bit DAC with an output range of + 10V, four 16-bit ADCs and four 12-bit ADCs with an input range of + 10V. Moreover, it comprises two channels for incremental encoder (IE) interface, used in this research study for rotational speed measurement.

In this research study, the dSPACE CLP1104 I/O board (connector panel) is used. It is used to connect the external devices to the board easily. The connector panel is connected to the DS1104 R&D Controller Board that is inserted into the PCI slot of the PC. The incremental encoder's output, which is connected to the dc motor's shaft, was connected to the Inc1 9-pin DSUB connector on the CLP 1104 I/O board. The Simulink Coder (previously called as Real-Time Workshop) is used to produce code of the model whereas RTI facilitates blocks that execute the I/O capabilities of the dSPACE system in the Simulink models. Henceforth, just by a single click, without requiring writing even a single line of code, the real-time model is automatically downloaded, executed, and started. The RTI guides during the configuration. Furthermore, the Control Desk Next Generation (CDNG), by dSPACE, is the development software for seamless Electronic Control Unit (ECU) realization. It executes all the essential tasks and provides one working environment, right from the beginning of the experimentation until the end. A control panel using dSPACE Control Desk Next Generation is designed (*.lay) which serves as a user-interface. The CDNG facilitates the visualization of the output parameters, like, reference rotor speed, the actual rotational speed, speed error, reference torque, torque and torque error.

To interact with the real-time which is running on the DS1104 R&D Controller Board, Control Desk Next Generation requires concurrent use of several files. Amongst the generated files, are the executable program (*.ppc), variable description file (*.sdf) used to relate to variables and the parameters in the model, virtual instrument panels i.e. layouts (*.lay) etc.

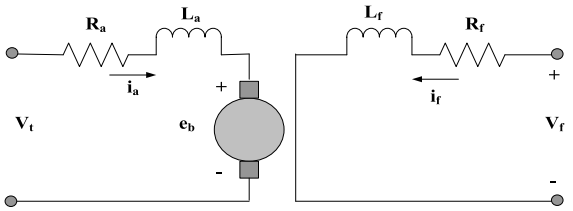

FIGURE 3. Separately excited DC motor.

B. IMPLEMENTATION OF SIMULINK MODEL ON TEST-BENCH

The MATLAB/Simulink development environment is used for the design and modeling of the wind turbine, as well as for the design of digital PI controllers. Besides producing the reference speed, another essential purpose is to generate PWM signals for triggering the gates of IGBT modules. Moreover, the I/O ports of the dSPACE CLP 1104 connector panel are easily accessible from inside the Simulink library browser. Upon building the Simulink control system, using the real-time option by pressing CTRL + B, the entire system is executed inside the dSPACE board, DS1104 R&D Controller Board. It indicates that the control system, which was previously implemented in software (Simulink), has been changed to operate in real-time on a hardware board. Upon building the Simulink control system *.sdf file is generated. This generated file provides the user access to the variables of the control system (e.g. tuning gain of the controllers and rotational speed etc.) to separate software called Control Desk Next Generation. In this software a control panel is created which offers the possibility of changing the variables of the control system in real-time to communicate with the DS1104 R&D Controller Board and thus offers to vary the input parameters of the system. Consequently, the input parameters of WECS such as wind speed, rotor size, air density, and Cp can be changed easily. In this way, a real-time application is implemented on the dSPACE hardware.

III. MODELING OF PROPOSED WIND TURBINE EMULATOR

In the following subsections, a thorough description of the modeling of dc motor and wind turbine emulator is provided.

A. MODELING OF THE DC MOTOR

In the proposed WTE, the separately excited dc motor replaces the HAWT. Figure 3 depicts the equivalent circuit for a separately excited dc motor. The speed control is performed within the base speed of the motor (i.e. 1750 rpm). In other words, only the constant torque region is considered, with no flux weakening. The following Equation 1 describes the dc motor's armature voltage, where e_b is the back EMF voltage given by Equation 2. The separately excited field voltage (V_f) is given by Equation 3. The dc motor torque is given by Equation 4, while Equation 5 presents the dynamic

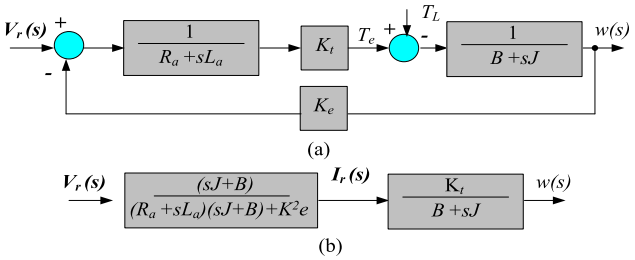

FIGURE 4. DC Motor modeling (a) Block diagram of the DC motor, (b) DC motor block diagram after the block diagram reduction.

TABLE 1. The parameters of the dc motor.

Parameter	Symbol	Value
Rated speed	ω	1750 r.p.m (183.256 rad/sec)
Rated power	P	$\frac{1}{3}$ h.p (248.6666 W)
Rated voltage	V	115 V
Rated current	I_a	3.2 A
Armature resistance	R_a	5 Ω
Armature inductance	L_a	175 x 10^{-3} H
Moment of inertia	J	9.07 x 10^{-3} kg.m^2
Coefficient of viscous friction	B	8.0 x 10^{-3} kg.m^2/sec
Back e.m.f constant	K_e	0.333 V/(rad/sec)
Torque constant	K_t	0.333 Nm/A

torque-speed characteristic of the motor.

$$V_t = R_a i_a + L_a \frac{di_a}{dt} + e_b \tag{1}$$

$$e_b = K_e \omega \tag{2}$$

$$V_f = R_f i_f + L_f \frac{di_f}{dt} \tag{3}$$

$$T_m = K_t i_a \tag{4}$$

$$T_e = T_l + J \frac{d\omega}{dt} \tag{5}$$

K_e and K_t are constants, which in actuality depend on the flux per pole of the dc machine. However, since the control only considers the constant torque region (with no flux weakening), the flux per pole is treated as a constant. Based on Equations 1 to 5, a block diagram representation of the dc motor can be constructed, as shown in Figure 4 (a). In this diagram, it is assumed that $T_l = B\omega + T_L$ where B stands for viscous friction and T_L is the speed independent load component. Using the block diagram transformation technique, the dc machine block diagram can be transformed into Figure 4 (b). It can be seen that the transfer function between the armature voltage, V_t, and the armature current, I_a (or torque, T_e), contains zero and two poles, which are determined by the parameters of the machine.

The rated speed (ω), rated power (P), rated voltage (V), rated current (I_a) are usually given on the nameplate of the machine. Some parameters like armature resistance (R_a), armature inductance (L_a), moment of inertia (J), coefficient of viscous friction (B), back e.m.f constant (K_e) and torque constant (K_t) need to be extracted by performing tests and measurements onto the dc machine. The extracted parameters obtained after performing these tests and measurements are presented in Table 1.

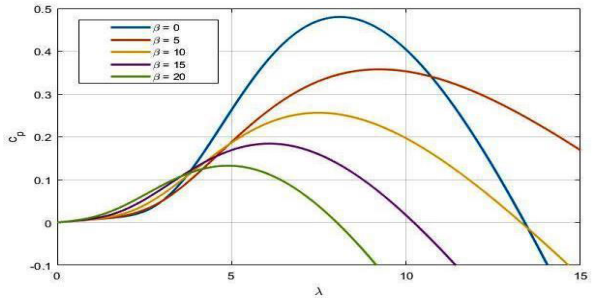

FIGURE 5. Cp vs λ curve.

IV. WIND TURBINE CHARACTERISTICS AND MATHEMATICAL MODELING

The turbine's mechanical power is given by Equation 6, where, πR^2 is the rotor swept area; therefore, Equation 6 can be modified as Equation 7, P_m is the mechanical power, ρ denotes air density, R represents the rotor radius, power coefficient is C_p, v is the wind speed, pitch angle is β and λ is tip speed ratio, which is given by Equation 8. The C_p is a measure of the wind turbines conversion efficiency. It is commonly known as Betz's constant and is a function of (λ, β). Theoretically, 59.3% of the total amount of wind power available can be extracted [70]. However, in actuality, the wind turbine installed at the site can only convert approximately 35-45% of the available wind power. The power coefficient, C_p is given by Equation (9). Here, $\frac{1}{\lambda_i} = \frac{1}{\lambda+0.008\beta} - \frac{0.035}{\beta^3+1}$, and with empirical constants as, $C_1 = 0.5176$, $C_2 = 116$, $C_3 = 0.4$, $C_4 = 5$, $C_5 = 21$ and $C_6 = 0.0068$.

The curve for C_p vs λ is shown in Figure 5 for different values of pitch angle (β), at wind speed of 8 m/s, obtained from the developed WTE.

Figure 6 shows the mechanical power vs rotor speed, for a fixed wind speed of 8 m/s with zero pitch angle i.e. $\beta = 0$, also obtained from the developed WTE.

$$P_m = 0.5\rho\pi R^2 C_p(\lambda, \beta)v^3 \tag{6}$$

$$P_m = 0.5\rho A C_p(\lambda, \beta)v^3 \tag{7}$$

$$\lambda = \frac{\omega R}{\upsilon} \tag{8}$$

$$C_p = C_1\left(\frac{C_2}{\lambda_i} - C_3\beta - C_4\right)e^{-C_5/\lambda_i} + C_6\lambda \tag{9}$$

$$T_{tur} = 0.5\rho\pi R^3 C_t(\beta, \lambda)v^2 \tag{10}$$

$$C_t = \frac{C_p}{\lambda} \tag{11}$$

Figure 6 demonstrates that the maximum mechanical power is 115 watts which occurs at the speed of 40 rad/sec and a corresponding maximum C_p magnitude of 0.45. The turbine torque is expressed in Equation 10. In Equation10 C_t denotes torque coefficient, which is given in Equation 11. The dynamic torque equation of the wind turbine is given by Equation 12, where T_{tur} denotes the turbine torque and T_{gen} is the torque of the generator, J_{tur} and B_{tur} are the moment of inertia and viscous friction constant of the turbine, respectively. In modeling the WTE, the rotational speed of the

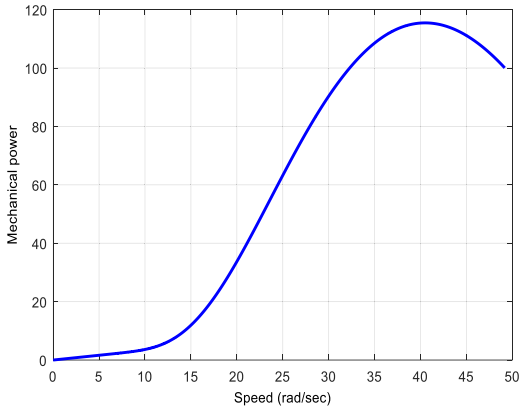

FIGURE 6. Power vs rotor speed characteristics.

wind turbine is obtained from Equation 13, which is written in the frequency domain.

To precisely replicate the features of wind turbines; the current research work utilizes the above mentioned WECS equations. In this study, β is kept constant at 0^o. And it is assumed that the speed of the generator is controlled such that the wind turbine is operated at its optimum capacity, i.e., $C_p = 0.45$.

$$T_{tur} - T_{gen} = J_{tur}\frac{d\omega_{tur}}{dt} + B_{tur}\omega_{tur} \tag{12}$$

$$\omega_{tur} = \frac{0.5\rho\pi R^3 C_t v^2 - T_{gen}}{sJ_{tur} + B_{tur}} \tag{13}$$

A. THE PROPOSED STRUCTURE OF CONTROL SYSTEM FOR WIND TURBINE EMULATOR

As mentioned earlier, the dc motor is used to emulate the speed and the torque of the wind turbine, which is based on the mathematical model of the wind turbine. The closed-loop dc motor control system uses the wind turbine speed reference that is generated by the mathematical model of the wind turbine. To implement this speed reference, a cascade control structure for the dc motor closed-loop system is adopted, as shown in Figure 7.

V. DESIGN OF CONTROLLERS FOR WIND TURBINE EMULATOR

The cascaded control system structure consists of two control loops. The outer loop is the speed control loop, whereas, the inner is the torque control loop. It is essential to ensure the speed of the dc motor precisely follows the reference speed produced by the wind turbine model. Since, in this study two types of controllers proposed, designed and implemented, i.e., PI controllers and Hybrid controller (ANFIS-PI). In the PI control structure, both the control loops, i.e. outer loop (speed controller) and inner loop (torque controller) are based on PI controllers. Whereas, the Hybrid control scheme is a combination of ANFIS and PI controller. In a hybrid control scheme, the outer controller (speed controller) is an ANFIS whereas the inner (torque controller) is a PI. A performance comparison between these two controllers is presented in this paper. The PI controllers' design procedure is presented first, and then the HC.

A. PI CONTROLLERS

The PI controllers for the torque and speed loops are designed based on the system's small-signal model. The small-signal model for the dc machine is obtained based on Equations 1 to 5, as represented by the block diagram, as shown in Figure 4. Since the pulse width modulated signals are generated based on a comparison between the triangular wave and a control signal, it is well known that the small-signal model of the converter is essentially presented by just a dc gain given by Equation 14.

$$G(s)_{conv} = \frac{V_{dc}}{2\hat{V}_{tri}} \tag{14}$$

In this study, the Vdc = 70 V and $\hat{V}_{tri}$ = 5 V, so that $G(s)_{conv} = 7$.

The design of the PI controllers is based on Bode plots. To ensure that the motor speed precisely follows the reference speed (generated by the wind turbine model), the response should be the fastest possible; this can be achieved by ensuring that the closed-loop bandwidth is as large as possible. Also, steady-state error should be zero; which can be achieved by ensuring that the dc gain of the open-loop transfer function is infinite or very large. As well as, good stability; using Bode plots, good stability translates to a good phase margin, typically $> 65^o$.

B. TORQUE CONTROLLER

The design starts with the innermost loop, i.e., torque loop, and proceeds towards the outer loop, i.e. speed loop. The small-signal model of the inner torque loop with the PI controller is shown in Figure 8.

In this work, the switching frequency of the converter, f_{sw}, is selected to be 5 kHz. Therefore, the respective torque and speed bandwidths are as follows:

$$(BW)_{torque} = 3141.5\ rad/s$$
$$(BW)_{speed} = 31.415\ rad/s$$

The open-loop Bode plot for the uncompensated torque loop (i.e. without PI controller) is shown by the blue traces in Figure 9. From the plot, it can be seen that the control requirements are not fulfilled. Response time will be too long, and steady-state errors will be significant. When the loop gain is used with the PI controller, the values of the $k_{p,T}$, and $k_{i,T}$ for the PI controller have to be selected so that the control requirements, mentioned above, i.e., (fast response, zero steady state error, good stability) are fulfilled for the torque loop. The torque's closed-loop bandwidth can be approximated by the crossover frequency of the open-loop transfer function, which can be obtained from the Bode plot.

Thus suitable values of the controllers can be selected to achieve this. The selection of $k_{p,T}$, and $k_{i,T}$ is simplified by separating the PI controller into three terms:

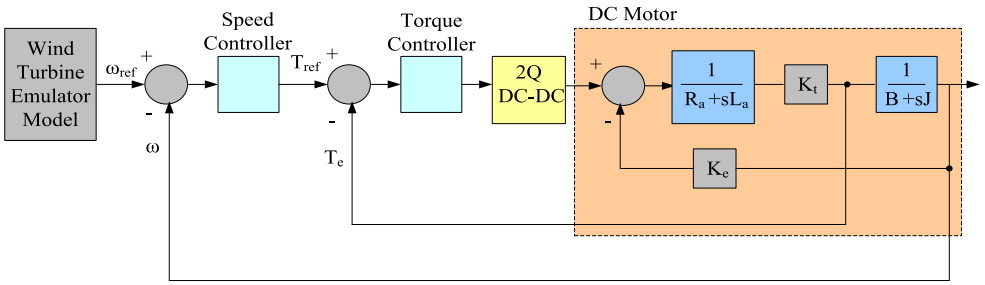

FIGURE 7. Cascade control of DC motor for WTE.

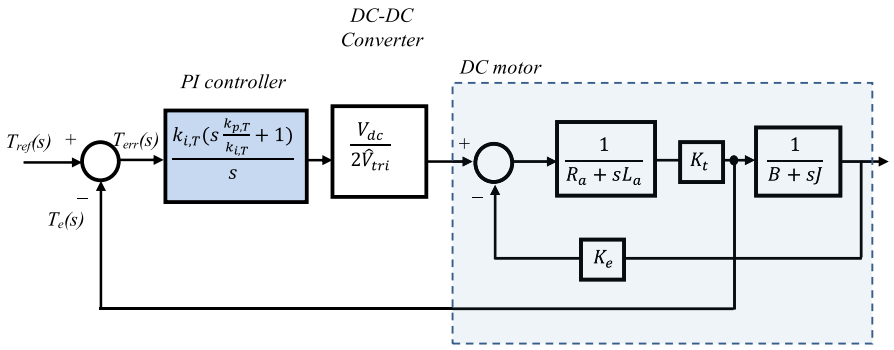

FIGURE 8. Small-signal model for torque loop.

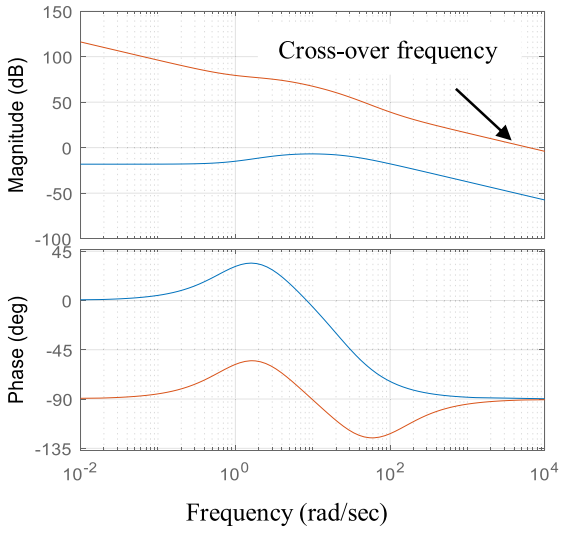

FIGURE 9. The bode plot for torque loop.

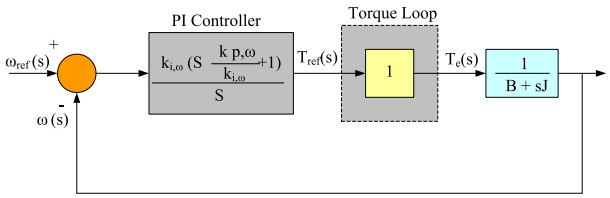

FIGURE 10. Small-signal model for speed loop.

The zero location of the PI controller is selected to be in between the poles of the dc motor. By locking the location of the zero, the DC gain can be conveniently adjusted to obtain the desired crossover frequency without affecting the phase. These values are selected by using the Control System Designer tool available in SIMULINK.

TABLE 2. The coefficients of PI torque controller.

PI Torque controller	
$k_{p,T}$	38
$k_{i,T}$	3400

TABLE 3. The coefficients of PI speed controller.

PI Speed Controller	
$k_{p,\omega}$	0.142
$k_{i,\omega}$	0.252

The final values of the $k_{p,T}$ and $k_{i,T}$ for the torque PI controller are listed in Table 2 and shown by the corresponding Bode plot with orange traces in Figure 9 with Eq. 15. As can be seen from the plots, the crossover frequency is about 3141.5 rad/s, and the dc gain is infinite. Furthermore, the phase margin is larger than 65°, as required.

$$20\log_{10}\left\{\frac{k_{i,T}\left(s\frac{k_{p,T}}{k_{i,T}}+1\right)}{s}\right\} = 20\log_{10}k_{i,T} + 20\log_{10}\times\left(s\frac{k_{p,T}}{k_{i,T}}+1\right) + 20\log_{10}\left(\frac{1}{s}\right) \tag{15}$$

C. PI SPEED CONTROLLER

Since $(BW)_{speed}$ is a decade lower than $(BW)_{torque}$, therefore, the torque closed-loop gain can be assumed to be unity

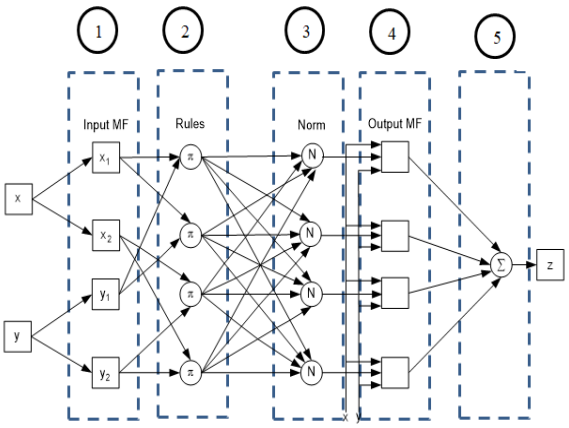

FIGURE 11. ANFIS structure.

when designing the speed controller. In this way, the viscous friction constant (B), and the system's moment of inertia (J), are primarily responsible for the proportional and integral gains of the speed PI controller, $k_{p,\omega}$, and $k_{i,\omega}$ respectively; the small-signal model of the speed loop with the PI controller is shown in Figure 10.

Since the system contains only one pole the zero location of the PI is therefore conveniently selected to cancel this. Again, by using the Control System Designer tool, suitable values of $k_{p,\omega}$, and $k_{i,\omega}$ are chosen such that the cross over frequency, i.e. the $(BW)_{speed}$, is at 31.415 rad/s. The final values $k_{p,\omega}$, and $k_{i,\omega}$ are listed in Table 3.

D. ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM CONTROLLERS

ANFIS is a tool that utilizes neural properties to understand the logic of the data set and utilizes the fuzzy methodology to apply its logic further effectively over the data set. ANFIS is a very powerful method and deals with the modeling of nonlinear and complex systems. ANFIS consolidates the benefits of ANN ability to gain information from data processing. It also uses FLC property for dealing with uncertainties in data. Due to this, ANFIS can approximate nonlinearity and uncertainties in a system irrespective of the numerical analysis. It has both the advantages of FLC and ANN. The ANFIS uses Tekagi-Sugeno fuzzy inference system. In terms of self-learning, parallel processing, adoption, and generalization abilities, of neuro-fuzzy systems it has certain benefits. ANFIS has been effectively used to address various engineering problem. ANFIS is the most successful method for optimizing the Membership Functions (MFs) of the Fuzzy Inference System (FIS) to minimize the error given that it has characteristics like a single output produced using weighted average defuzzification. It has no rule sharing and one weight for each rule.

E. STRUCTURE OF THE ANFIS CONTROLLER

ANFIS consists of five layers (labeled 1-5); each layer is connected through weights, as shown in Figure 11.

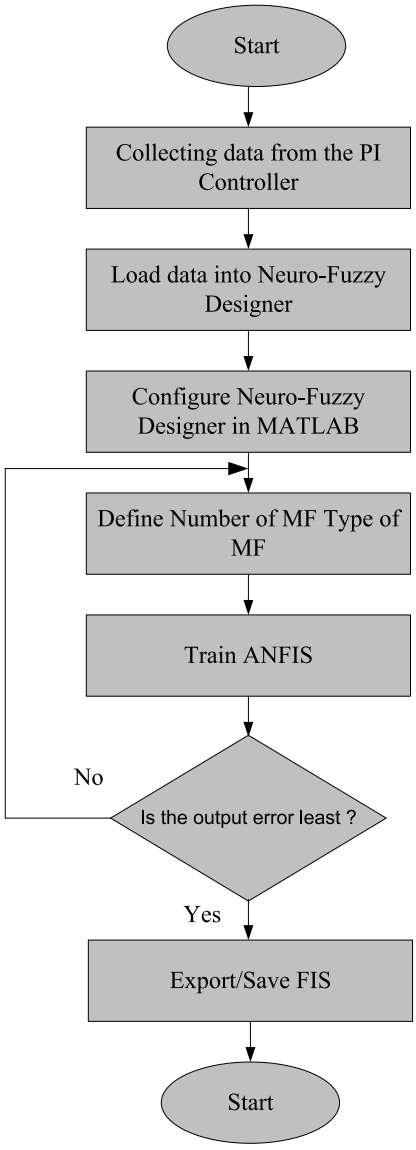

FIGURE 12. Flowchart of designing ANFIS controller.

In the first layer (input layer), the received input data is mapped into MFs to determine the membership degree of that input. The x and y denote input to the ANFIS. The x_1, x_2 are the input MFs of x. Similarly, y_1, y_2 are the input MFs of y. In the second layer, fuzzy rules are used to develop a relationship between input and output. The third layer normalizes the output and passes it to the fourth layer. The fourth layer maps the output data and provides the output MF. In the fifth layer, the outputs are added to provide a single output. It is important to mention here that ANFIS can deal with multiple inputs. However, in this research study only one input is used.

F. DESIGN OF ANFIS CONTROLLERS

The flowchart for the ANFIS controller design is shown in Figure 12. Before the controller can be used, it needs to be trained and optimized. The design of the ANFIS controller is performed with the help of MATLAB toolbox. The pro-

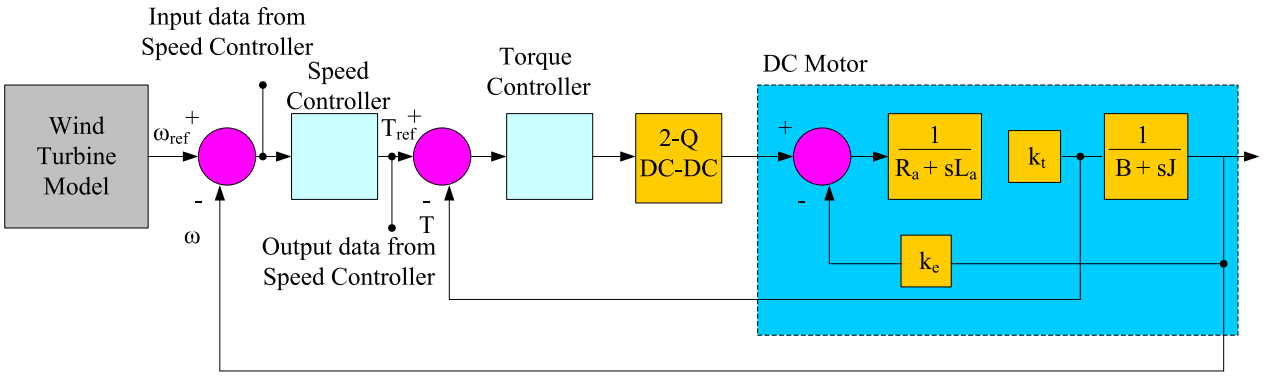

FIGURE 13. Extracting data for training of the anfis from the pi-based control system.

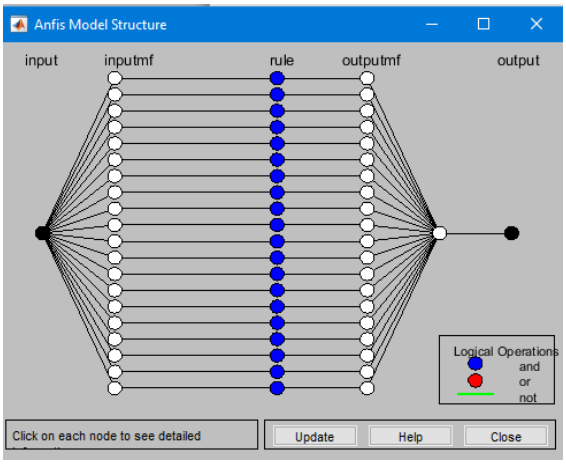

FIGURE 14. ANFIS model structure.

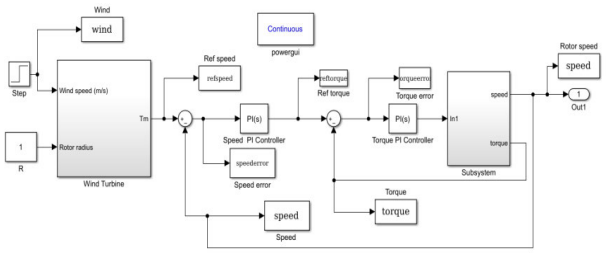

FIGURE 15. Simulink model for PI controllers with WTE model.

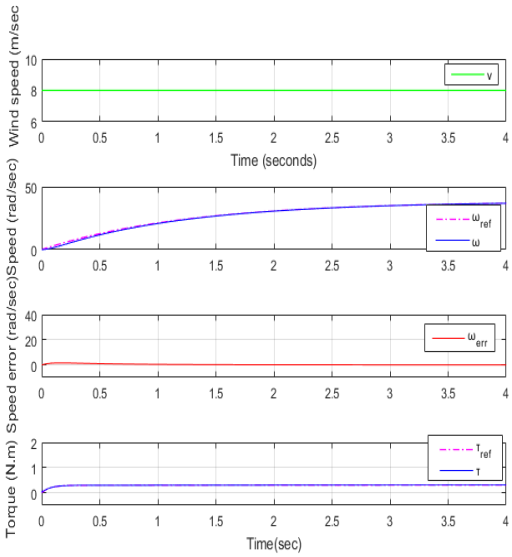

FIGURE 16. Simulation results based on PI controllers with WTE model.

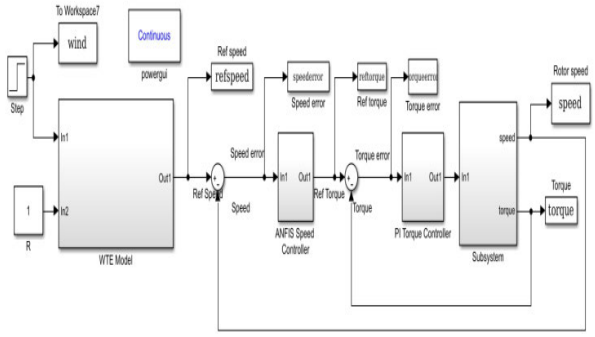

FIGURE 17. Simulink model for hybrid controller with WTE model.

cess starts with collecting data for training purposes. In this work, the data is obtained by running experiments on the PI controllers based control system, which has been designed. For the purpose of collecting data, a step-change in the speed reference is used. The data is collected using the Control Desk software from dSPACE, which is then saved as a.mat file (compatible with MATLAB).

A total of 166,645 pairs of input-output data for speed PI controller are recorded. It should be noted that higher the data better will be the accuracy. However, it will take more time for training. The point at which the input-output data is collected is shown in Figure 13. The input for the speed training data is obtained from the speed error, nevertheless, during training the error was found to be 0.0537.

Next, after the data is collected, the training process is performed with the help of the Neuro-Fuzzy Designer tool that can be opened from the apps tab or by simply typing anfisedit at the MATLAB command prompt. Using Neuro-Fuzzy Designer, one can also select the number and type of MF to be used for the input and output layers of the ANFIS. In this work, 20 triangular MFs are chosen for each input and output. Triangular MFs are selected due to their simplicity

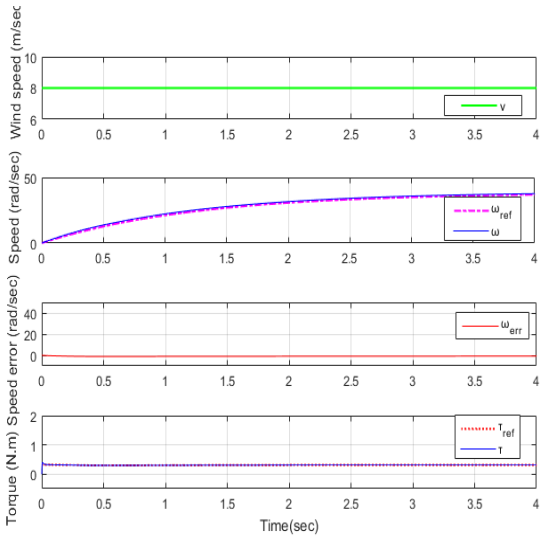

FIGURE 18. Results based on hybrid controller with WTE model.

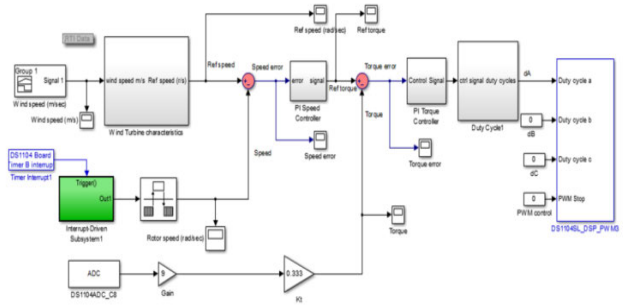

FIGURE 19. Simulink model based on PI controllers.

and require less training and execution time. The training data is then loaded to the ANFIS editor. In this work, for the training configuration, the hybrid optimization approach is chosen. Back propagation and least-squares techniques are combined in this optimization technique and gives the highest accuracy.

Further, the error tolerance is set as zero. The number of epochs is chosen as 100 to achieve suitable training accuracy and time. In this work, the training is stopped when the number of epochs is completed, i.e. 100.

When the training process is completed, and the pre-set error tolerance is approximately achieved, the generated fis (fuzzy inference system) is exported to the MATLAB Workspace. The designed controller can be linked to the Simulink model using the Fuzzy Logic Controller block, which is then configured using the generated.fis file. Finally, the structure of the developed ANFIS controller can also be viewed by clicking the Structure button in the Neuro-Fuzzy Designer window, as shown in Figure 14.

VI. RESULTS AND DISCUSSION

The maximum overshoot, rise time and settling time are the key parameters for evaluating the performance of any

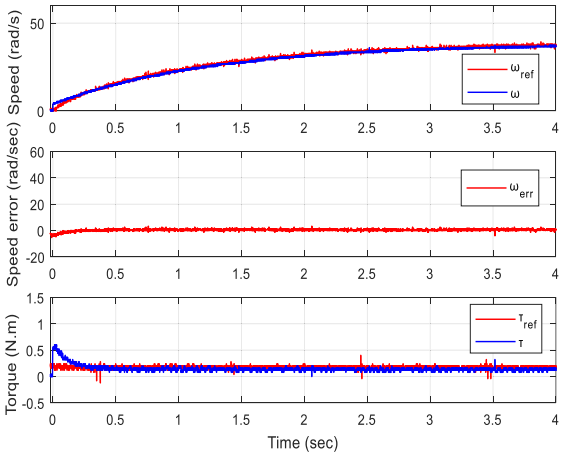

FIGURE 20. Results based on PI controllers with WTE model.

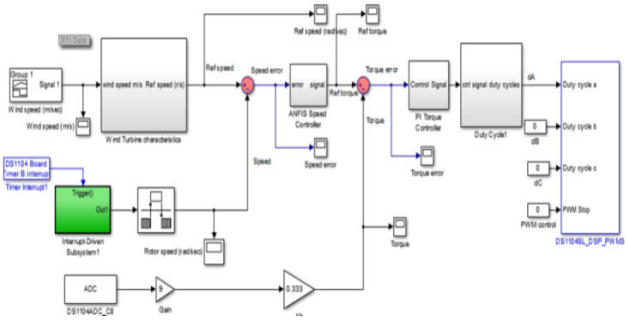

FIGURE 21. Simulink model based on hybrid control scheme.

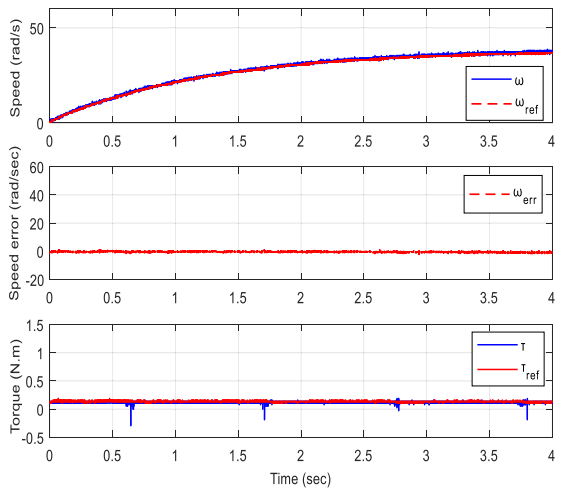

FIGURE 22. Results based on hybrid controller.

controllers. However, they are suitable for any step response evaluation. Therefore, before implementing the HC in the WTE, the performance of the proposed HC was evaluated using these mentioned parameters. A reference speed of 40 rad/sec was applied. The HC provided outstanding performance. However, in this paper the RMSE calculation is preferred considering the non-linearity of the system, i.e., instead of step change a random test signal was applied for evaluating the performance of the proposed HC.

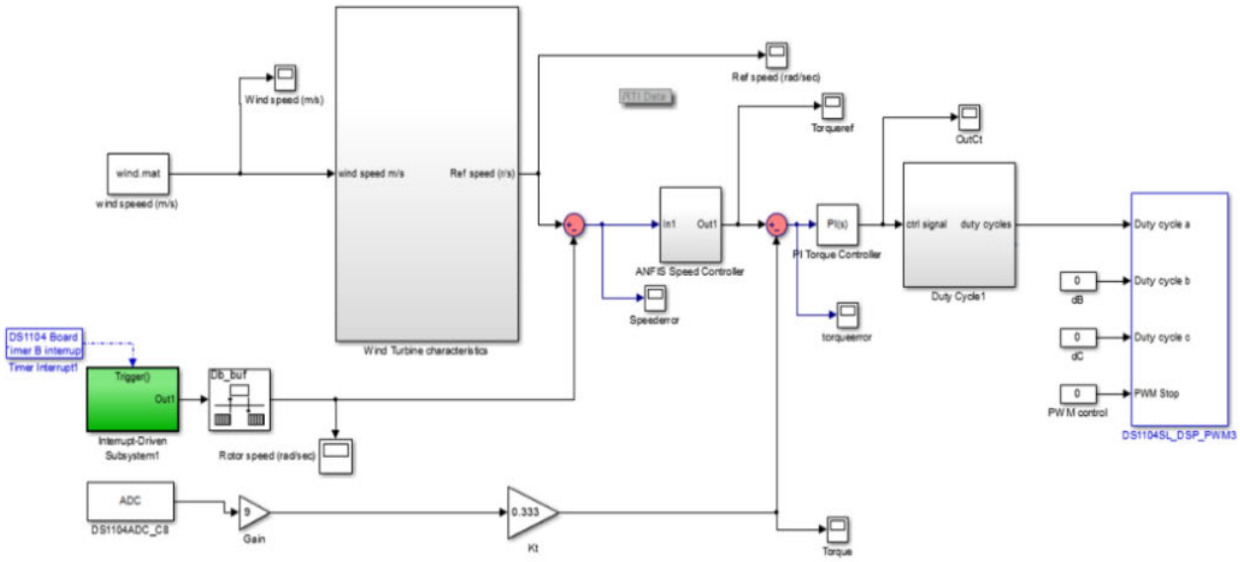

FIGURE 23. Simulink model used for testing WTE with random wind speed.

Therefore, RMSE calculation was suitable option for performance evaluation.

The results are categorized as simulation and experimental. First, simulation results are presented followed by the experimental results.

A. SIMULATION RESULTS

The behavior of the WTE is analyzed based on these two control schemes i.e., PI and HC scheme. These both control schemes are employed for the cascaded DC motor drive systems and for each scheme a step wind speed (0 m/s to 8 m/s) is fed to the wind turbine model. The block diagram for the PI control scheme is shown in Figure 15.The corresponding obtained results are shown in Figure 16.

From Figure 16, it can be seen that the actual motor speed managed to follow the reference speed, which is the generated wind turbine speed, with some initial speed error. The actual speed managed to track the increasing speed with minimum error. From the simulation results, the RMSE is calculated as 0.4940.

For HC control scheme, the structure of the control block diagram is similar to the PI controller scheme as before (Fig 15). However, the PI controller for the speed is now replaced with the ANFIS controller. The Simulink model for the HC is shown in Figure 17, while its corresponding speed and torque responses are shown in Figure 18. It can be seen from Figure 18 that the initial speed error is zero and the speed tracks the reference accurately.

From the simulation, the calculated RMSE, for HC scheme is found to be 0.1584. Clearly, the HC control scheme outperforms the PI controller with significantly lower RMSE value.

B. EXPERIMENTAL RESULTS

Two control schemes PI and HC are evaluated with wind turbine emulator generating the wind turbine speed in real-time. The wind turbine is simulated in real-time using the dSPACE DS1104 controller board, whereby the parameters of the wind turbine are exactly the same as the ones used in the simulation. Again, the input to the wind turbine is a step change in the wind speed, from 0 m/s to 8 m/s. The two control schemes were implemented using the same control board, i.e. DS1104. The actual speed is measured using an incremental encoder with 2048 p.p.r., whereas the armature current (torque) is measured using a Hall affect sensor. The signals for the speed (reference and actual), speed error and torque (reference and actual) are recorded and saved using a digital oscilloscope. These signals are then plotted using MATLAB. Similar to the simulation results, the RMSE is calculated to evaluate the performance of the WTE for the two different control schemes. The Simulink model for the proposed WTE is shown in Figure19 and the corresponding speed and torque characteristic curves are shown in Figure 20. As can be seen from Fig 20, there is an initial speed error, which is similar to what is obtained in the simulation results. As the speed builds up, the speed error reduces. The speed RMSE is calculated based on the speed error graph for 4 sec and the value is found to be equal to 0.7273. In HC scheme, the outer control loop uses the ANFIS controller while the inner control loop uses the PI controller. The Simulink model used to implement this control scheme is shown in Figure 21.

Unlike the PI based control scheme presented earlier, with Hybrid control scheme, the initial speed error is almost zero. Also, the motor speed managed to track the reference speed perfectly as the speed builds up. From the comparison between the two control schemes (PI based and Hybrid), the best control scheme in terms of speed response and speed RMSE is Hybrid. Therefore, for the study using the random wind speed and actual HAWT parameters, only Hybrid control scheme is considered. Based on the RMSE evaluation, it is observed that Hybrid control scheme has the lesser RMSE

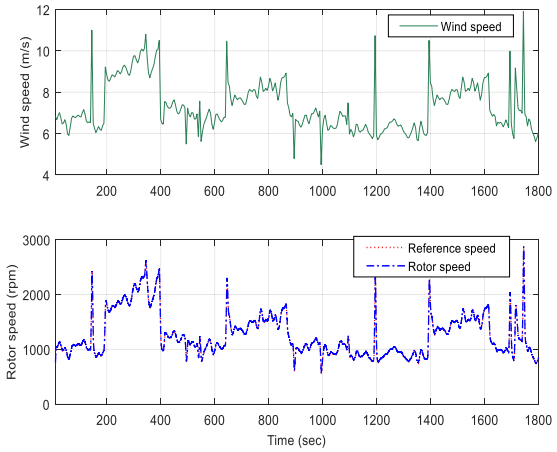

FIGURE 24. Random wind speed generator (a) Wind speed (b) Speed of wind turbine using the HAWT parameters in table 5.

TABLE 4. Parameters of HAWT for validation purpose.

Parameter	Symbol	Value
Rated speed	ω	180 rad/sec
Rated power	P_{tur}	Less than 3 kW
Rotor radius	R	1 m
Air density	ρ	1.125 kg/m^2
Moment of inertia	J_{tur}	0.3 kg.m^2
Coefficient of viscous friction	B_{tur}	0.24 kg.m^2/sec

value i.e., 0.495 when compared with the PI control scheme i.e., 0.7273.

C. TESTING OF PROPOSED WTE PERFORMANCE WITH RANDOM WIND SPEED

To evaluate the performance of the developed WTE, the HAWT parameters extracted from reference [3] are selected. The parameters of the HAWT obtained from [3], used for the WTE are listed in Table 4. These parameters are scaled down from the commercially available wind turbine for experimental implementation purposes. Instead of using a step change in wind speed, a random wind speed generated using MATLAB is used as the input to the emulator. The results obtained from the experiment can be directly compared with the results obtained from the literature [3], thus verifying the validity of the developed WTE.

Based on this wind speed and the parameters in Table 4, the wind turbine model generated the wind turbine speed (i.e. reference speed to the DC motor drive) as shown in Figure 25 (b), which is comparable to a typical speed of a turbine based on the actual wind speed extracted. The random wind speed is generated over 1800 seconds. The variation of the generated wind speed is limited between 4 m/s to 12 m/s, as shown in Figure 24.

The reference speed and the emulated wind turbine speed response obtained from the WTE using the random wind speed are shown in Figure 25 (b). The gear ratio was considered as four. Furthermore, to compare the current outcomes with that of the selected literature, the rotational speed is taken in rpm instead of rad/sec. As expected, the motor

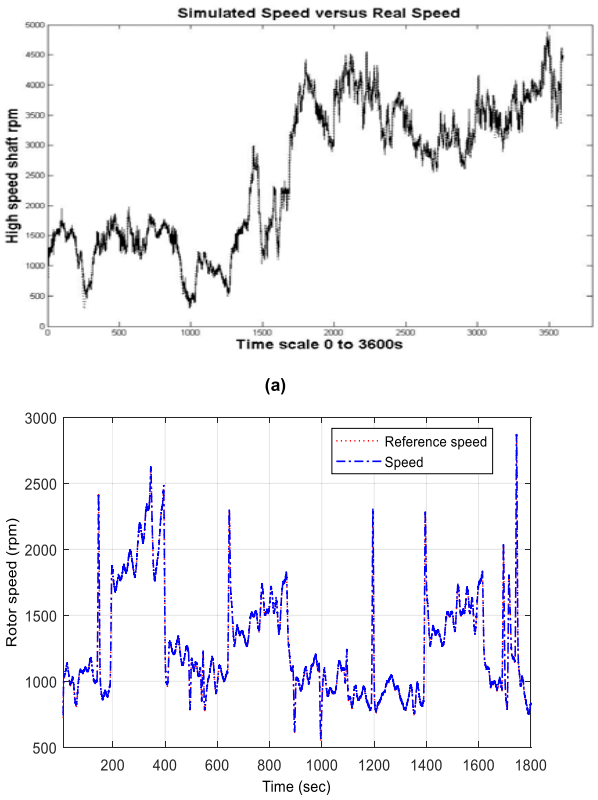

FIGURE 25. The behavior of WTE controllers to a random reference speed (a) selected literature [21] (b) proposed WTE.

speed managed to accurately emulate the speed of the wind turbine by producing speed that exactly follows the reference speed generated by the wind turbine model and calculated RMSE is found to be 0.5004.

The Simulink model used for testing the proposed WTE with random wind speed is shown in Figure 23, which has a similar structure as the one shown in Figure 21 except that wind speed is a random signal. The rand wind speed signal generated is shown in Figure 25(a). The performance of the proposed HC was compared in simulation as well as proved experimentally. Hence, zero percentage overshoot and the fastest response (0.081 sec) sound suitable for implementing the HC for WTE controllers.

VII. CONCLUSION

The use of PI controllers in WTE has significant drawbacks i.e., the overshoot and slow response. By optimally designing PI controllers, the overshoot can be improved nevertheless cannot be completely suppressed with fast response. The HC control structure which is the combination of PI and ANFIS proposed in this research study magnificently improved the overshoot and concurrently minimized settling time. This was made possible by proper training and implementing a trained fuzzy inference system by replacing the PI (outer) controller in cascaded control structure. It is also witnessed that when implemented to the WTE, with a step-change in

wind speed, the Hybrid control scheme produces the best result.The Root-Mean-Square Error (RMSE) for the speed of the Hybrid controller is found to be 0.495, which is lesser when compared to the PI control scheme i.e., 0.7273.

AUTHOR CONTRIBUTIONS

The authors contributed to the design and implementation of the research, to analysis the results and to write the manuscript. They read and approved the final version of the manuscript.

CONFLICTS OF INTEREST

The authors declare no competing interests.

ACKNOWLEDGMENT

This article outlines the result of a collaborative work conducted at Beijing Jiaotong University, Beijing, China, and Universiti Teknologi Malaysia, Johor Bahru, Malaysia.

REFERENCES

[1] M. Diaz, R. Cárdenas, M. Espinoza, A. Mora, and P. Wheeler, "Modelling and control of the modular multilevel matrix converter and its application to wind energy conversion systems," in *Proc. 42nd Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2016, pp. 5052–5057.

[2] M. A. Bhayo, M. J. A. Aziz, N. R. N. Idris, and A. H. M. Yatim, "Design and development of a wind turbine emulator for analyzing the performance of stand-alone wind energy conversion system," *Int. J. Power Electron. Drive Syst.*, vol. 8, pp. 454–461, Mar. 2017.

[3] L. Peretti, V. Särkimäki, and J. Faber, "A wind turbine emulator for generator control algorithm development," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Feb. 2013, pp. 228–233.

[4] M. Suwan, T. Neumann, C. Feltes, and I. Erlich, "Educational experimental rig for doubly-fed induction generator based wind turbine," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, Jul. 2012, pp. 1–8.

[5] N. Horiuchi and T. Kawahito, "Torque and power limitations of variable speed wind turbines using pitch control and generator power control," in *Proc. Power Eng. Soc. Summer Meeting. Conf.*, 2001, pp. 638–643.

[6] M. Arifujjaman, M. Iqbal, and J. Quacioe, "Development of an isolated small wind turbine emulator," *Open Renew. Energy J.*, vol. 4, no. 1, pp. 3–12, Jul. 2011.

[7] D. Dolan and P. Lehn, "Real-time wind turbine emulator suitable for power quality and dynamic control studies," M.S. thesis, Univ. Toronto, Toronto, ON, USA, 2005.

[8] L. Chang, R. Doraiswami, T. Boutot, and H. Kojabadi, "Development of a wind turbine simulator for wind energy conversion systems," in *Proc. Can. Conf. Elect. Comput. Eng.*, Halifax, NS, Canada, 2000, pp. 550–554.

[9] P. S. Kumar, R. P. S. Chandrasena, and K. V. Sam Moses Babu, "Design and implementation of wind turbine emulator using FPGA for stand-alone applications," *Int. J. Ambient Energy*, vol. 43, no. 1, pp. 2397–2409, Dec. 2022.

[10] T. Hardy and W. Jewell, "Emulation of a 1.5 MW wind turbine with a DC motor," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, Jul. 2011, pp. 1–8.

[11] M. Monfared, H. M. Kojabadi, and H. Rastegar, "Static and dynamic wind turbine simulator using a converter controlled DC motor," *Renew. Energy*, vol. 33, no. 5, pp. 906–913, May 2008.

[12] M. Chinchilla, S. Amaltes, and J. L. Rodriguez-Amenedo, "Laboratory set-up for wind turbine emulation," in *Proc. IEEE Int. Conf. Ind. Technol.*, 2004, pp. 553–557.

[13] M. Arifujjaman, M. Iqbal, and J. E. Quaicoe, "Emulation of a small wind turbine system with a separately-excited DC machine," *IU-J. Electr. Electron. Eng.*, vol. 8, pp. 569–579, Jun. 2012.

[14] P. Ochieng, A. Manyonge, and A. Oduor, "Mathematical analysis of tip speed ratio of a wind turbine and its effects on power coefficient," *Int. J. Math. Soft Comput.*, vol. 4, no. 1, pp. 61–66, 2014.

[15] K. Han and G.-Z. Chen, "A novel control strategy of wind turbine MPPT implementation for direct-drive PMSG wind generation imitation platform," in *Proc. IEEE 6th Int. Power Electron. Motion Control Conf.*, May 2009, pp. 2255–2259.

[16] L. Y. Pao and K. E. Johnson, "A tutorial on the dynamics and control of wind turbines and wind farms," in *Proc. Amer. Control Conf.*, 2009, pp. 2076–2089.

[17] I. Szeidert, O. Prostean, I. Filip, C. Vasar, and L. Mihet-Popa, "Issues regarding the modeling and simulation of wind energy conversion system's components," in *Proc. IEEE Int. Conf. Autom., Quality Test., Robot.*, 2008, pp. 225–228.

[18] J. G. Slootweg, H. Polinder, and W. L. Kling, "Initialization of wind turbine models in power system dynamics simulations," in *Proc. IEEE Porto Power Tech.*, Sep. 2001, p. 6.

[19] C. I. Martínez-Márquez, J. D. Twizere-Bakunda, D. Lundback-Mompó, S. Orts-Grau, F. J. Gimeno-Sales, and S. Seguí-Chilet, "Small wind turbine emulator based on lambda-Cp curves obtained under real operating conditions," *Energies*, vol. 12, no. 13, p. 2456, Jun. 2019.

[20] G. J. Herbert, S. Iniyan, E. Sreevalsan, and S. Rajapandian, "A review of wind energy technologies," *Renew. Sustain. Energy Rev.*, vol. 11, no. 6, pp. 1117–1145, 2007.

[21] R. I. Ovando, J. Aguayo, and M. Cotorogea, "Emulation of a low power wind turbine with a DC motor in MATLAB/simulink," in *Proc. IEEE Power Electron. Spec. Conf.*, vols. 1–6, Aug. 2007, pp. 859–864.

[22] P. K. Banerjee and M. Arifujjaman, "Development of a test-rig for large scale wind turbine emulation," in *Proc. Int. Conf. Electr. Comput. Eng. (ICECE)*, Dec. 2010, pp. 159–162.

[23] B. Rabelo, W. Hofmann, and M. Gluck, "Emulation of the static and dynamic behaviour of a wind-turbine with a DC-machine drive," in *Proc. IEEE 35th Annu. Power Electron. Spec. Conf.*, Jun. 2004, pp. 2107–2112.

[24] F. E. V. Taveiros, L. S. Barros, and F. B. Costa, "Wind turbine torque-speed feature emulator using a DC motor," in *Proc. Brazilian Power Electron. Conf.*, Oct. 2013, pp. 480–486.

[25] G. Liu, S. Wang, and J. Zhang, "Design and realization of DC motor and drives based simulator for small wind turbine," in *Proc. Asia–Pacific Power Energy Eng. Conf.*, 2010, pp. 1–4.

[26] J. Hussain and M. K. Mishra, "Design and development of real-time small-scale wind turbine simulator," in *Proc. IEEE 6th India Int. Conf. Power Electron. (IICPE)*, Dec. 2014, pp. 1–5.

[27] W. Li, D. Xu, W. Zhang, and H. Ma, "Research on wind turbine emulation based on DC motor," in *Proc. 2nd IEEE Conf. Ind. Electron. Appl.*, May 2007, pp. 2589–2593.

[28] J. F. Mushi, K. Han, G. Chen, and J. Daozhuo, "Design and implementation of wind turbine imitation system for direct drive permanent magnet synchronous generator using DC motor," in *Proc. Int. Conf. Sustain. Power Gener. Supply*, 2009, pp. 1–6.

[29] S. T. Sager, M. A. Khan, and P. S. Barendse, "Laboratory setup of a grid-tied PM WECS for experimental investigation," in *Proc. IEEE Int. Energy Conf.*, Dec. 2010, pp. 524–529.

[30] K. Kariyawasam, K. Karunarathna, R. Karunarathne, M. Kularathne, and K. Hemapala, "Design and development of a wind turbine simulator using a separately excited DC motor," *Smart Grid Renew. Energy*, vol. 4, no. 3, pp. 259–265, 2013.

[31] A. K. Yadav, M. Singh, and D. C. Meena, "Modelling and simulation of wind turbine emulator using DC motor," in *Proc. IEEE 7th Power India Int. Conf. (PIICON)*, Nov. 2016, pp. 1–5.

[32] H. Garg and R. Dahiya, "Modeling and development of wind turbine emulator for the condition monitoring of wind turbine," *Int. J. Renew. Energy Res.*, vol. 5, pp. 591–597, 2015.

[33] I. Moussa, A. Bouallegue, and A. Khedher, "Design and implementation of constant wind speed turbine emulator using MATLAB/simulink and FPGA," in *Proc. 9th Int. Conf. Ecol. Vehicles Renew. Energies (EVER)*, Mar. 2014, pp. 1–8.

[34] B. Gong and D. Xu, "Real time wind turbine simulator for wind energy conversion system," in *Proc. IEEE Power Electron. Spec. Conf.*, Jun. 2008, pp. 1110–1114.

[35] F. A. Farret, R. Gules, and J. Marian, "Micro-turbine simulator based on speed and torque of a DC motor to drive actually loaded generators," in *Proc. 1st Int. Caracas Conf. Devices, Circuits Syst.*, 1995, pp. 89–93.

[36] M. Balaji, S. K. Sarangi, and M. Pattnaik, "Design of a DC motor based wind turbine emulator using sliding mode control approach," in *Proc. IEEE 1st Int. Conf. Energy, Syst. Inf. Process. (ICESIP)*, Jul. 2019, pp. 1–5.

[37] A. G. A.-K. Byunggyu Yu, "Wind turbine simulator development using a separately excited DC motor," *Int. J. Advancements Comput. Technol.*, vol. 5, no. 11, pp. 347–357, Jul. 2013.

[38] A. Abdelkafi, A. Masmoudi, and L. Krichen, "Experimental investigation on the performance of an autonomous wind energy conversion system," *Int. J. Electr. Power Energy Syst.*, vol. 44, no. 1, pp. 581–590, Jan. 2013.

[39] R. Patel, C. V. Patki, and V. Agarwal, "Armature and field controlled DC motor based wind turbine emulation for wind energy conversion systems operating over a wide range of wind velocity," in *Sustainability Energy Buildings*. Singapore: Springer, 2009, pp. 117–125.

[40] N. C. Sahoo, A. S. Satpathy, N. K. Kishore, and B. Venkatesh, "D.C. motor-based wind turbine emulator using LabVIEW for wind energy conversion system laboratory setup," *Int. J. Electr. Eng. Educ.*, vol. 50, no. 2, pp. 111–126, Apr. 2013.

[41] B. Neammanee, S. Sirisumrannukul, and S. Chatratana, "Development of a wind turbine simulator for wind generator testing," *Int. Energy J.*, vol. 8, pp. 21–28, 2007.

[42] L. A. Lopes, J. Lhuilier, M. F. Khokhar, and A. Mukherjee, "A wind turbine emulator that represents the dynamics of the wind turbine rotor and drive train," in *Proc. IEEE 36th Power Electron. Spec. Conf.*, 2005, pp. 2092–2097.

[43] W. Hu, Y. Wang, X. Song, and Z. Wang, "Development of wind turbine simulator for wind energy conversion systems based on permanent magnet synchronous motor," in *Proc. Int. Conf. Elect. Mach. Syst.*, 2008, pp. 2322–2326.

[44] L. Yang, S. Yan, Z. Chen, and W. Liu, "A novel wind turbine simulator for wind energy conversion systems using an permanent magnet synchronous motor," in *Proc. Int. Conf. Electr. Mach. Syst. (ICEMS)*, Busan, South Korea Oct. 2013, pp. 2156–2158.

[45] J. Chen, J. Chen, C. Gong, and H. Wang, "Design and analysis of dynamic wind turbine simulator for wind energy conversion system," in *Proc. 38th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2012, pp. 971–977.

[46] F.-J. Lin, L.-T. Teng, P.-H. Shieh, and Y.-F. Li, "Intelligent controlled-wind-turbine emulator and induction-generator system using RBFN," *IEE Proc.-Electr. Power Appl.*, vol. 153, no. 4, pp. 608–618, 2006.

[47] H. M. Kojabadi, L. Chang, and T. Boutot, "Development of a novel wind turbine simulator for wind energy conversion systems using an inverter-controlled induction motor," *IEEE Trans. Energy Convers.*, vol. 19, no. 3, pp. 547–552, Sep. 2004.

[48] M. A. Koondhar, M. Ali, M. U. Keerio, A. K. Junejo, I. A. Laghari, and S. Chandio, "Wind energy conversion system using maximum power point tracking technique—A comprehensive survey," *Appl. Eng. Lett. J. Eng. Appl. Sci.*, vol. 6, no. 4, pp. 148–156, 2021.

[49] J. M. Nye, J. G. De La Bat, M. A. Khan, and P. Barendse, "Design and implementation of a variable speed wind turbine emulator," in *Proc. 20th Int. Conf. Electr. Mach.*, Sep. 2012, pp. 2060–2065.

[50] H. Voltolini, M. H. Granza, J. Ivanqui, and R. Carlson, "Modeling and simulation of the wind turbine emulator using induction motor driven by torque control inverter," in *Proc. 10th IEEE/IAS Int. Conf. Ind. Appl.*, Nov. 2012, pp. 1–6.

[51] Z. H. Akpolat, G. M. Asher, and J. C. Clare, "Dynamic emulation of mechanical loads using a vector-controlled induction motor-generator set," *IEEE Trans. Ind. Electron.*, vol. 46, no. 2, pp. 370–379, Apr. 1999.

[52] M. Qiao, F. Lin, R. Hao, X. You, and T. Q. Zheng, "The research and development platform for wind energy system used induction motor replacing wind turbine," in *Proc. 2nd IEEE Conf. Ind. Electron. Appl.*, May 2007, pp. 2579–2582.

[53] H. M. Kojabadi and L. Chang, "A novel steady state wind turbine simulator using an inverter controlled induction motor," *Wind Eng.*, vol. 28, no. 4, pp. 433–443, Jun. 2004.

[54] K. Ohyama and T. Nakashima, "Wind turbine emulator using wind turbine model based on blade element momentum theory," in *Proc. SPEEDAM*, Jun. 2010, pp. 762–765.

[55] S. Mohod and M. Aware, "Laboratory development of wind turbine simulator using variable speed induction motor," *Int. J. Eng., Sci. Technol.*, vol. 3, no. 5, Aug. 2011.

[56] R. Teodorescu and F. Blaabjerg, "Flexible control of small wind turbines with grid failure detection operating in stand-alone and grid-connected mode," *IEEE Trans. Power Electron.*, vol. 19, no. 5, pp. 1323–1332, Sep. 2004.

[57] J. Castelló, J. M. Espí, and R. García-Gil, "Development details and performance assessment of a wind turbine emulator," *Renew. Energy*, vol. 86, pp. 848–857, Feb. 2016.

[58] I. Moussa and A. Khedher, "Software in-the-loop simulation of an advanced SVM technique for 2ϕ-inverter control fed a TPIM as wind turbine emulator," *Electronics*, vol. 11, no. 2, p. 187, Jan. 2022.

[59] A. Sajadi, Ł. Rosłaniec, M. Kłos, P. Biczel, and K. A. Loparo, "An emulator for fixed pitch wind turbine studies," *Renew. Energy*, vol. 87, pp. 391–402, Mar. 2016.

[60] I. Munteanu, A. I. Bratcu, S. Bacha, D. Roye, and J. Guiraud, "Hardware-in-the-loop-based simulator for a class of variable-speed wind energy conversion systems: Design and performance assessment," *IEEE Trans. Energy Convers.*, vol. 25, no. 2, pp. 564–576, Jun. 2010.

[61] H. Li, M. Steurer, K. L. Shi, S. Woodruff, and D. Zhang, "Development of a unified design, test, and research platform for wind energy systems based on hardware-in-the-loop real-time simulation," *IEEE Trans. Ind. Electron.*, vol. 53, no. 4, pp. 1144–1151, Jun. 2006.

[62] P. E. Battaiotto, R. J. Mantz, and P. F. Puleston, "A wind turbine emulator based on a dual DSP processor system," *Control Eng. Pract.*, vol. 4, no. 9, pp. 1261–1266, Sep. 1996.

[63] M. Arifujjaman, M. T. Iqbal, and J. Quaicoe, "An isolated small wind turbine emulator," in *Proc. Can. Conf. Electr. Comput. Eng.*, 2006, pp. 595–598.

[64] M. Arifujjaman, M. T. Iqbal, and J. E. Quaicoe, "Maximum power extraction from a small wind turbine emulator using a DC–DC converter controlled by a microcontroller," in *Proc. Int. Conf. Electr. Comput. Eng.*, Dec. 2006, pp. 213–216.

[65] L. Navarro, G. Christian, M. Quintero, and M. Pardo, "A lab-oriented testing platform for emulating a wind turbine in a DC microgrid," in *Proc. IEEE Texas Power Energy Conf. (TPEC)*, Feb. 2020, pp. 1–6.

[66] C. N. Bhende, S. Mishra, and S. G. Malla, "Permanent magnet synchronous generator-based standalone wind energy supply system," *IEEE Trans. Sustain. Energy*, vol. 2, no. 4, pp. 361–373, Oct. 2011.

[67] M. P. K. Bailapudi and N. Sinha, "Fuzzy logic controlled wind turbine emulator (WTE)," in *Proc. Int. Conf. Inf. Commun. Embedded Syst. (ICICES)*, Feb. 2016, pp. 1–8.

[68] L. Benaaouinate, M. Khafallah, A. Mesbahi, A. Martinez, and T. Bouragba, "Emulation of a wind turbine with a DC motor controlled by fuzzy logic controller," in *Proc. Int. Conf. Automat., Control Eng. Comput. Sci. (ACECS)*, vol. 20, 2017, pp. 97–101.

[69] J. C. Ferreira and L. G. B. Rlim, "Wind turbine emulator using an MPPT controller based on artificial neural network," in *Proc. 13th Brazilian Power Electron. Conf. 1st Southern Power Electron. Conf. (COBEP/SPEC)*, 2015, pp. 1–6.

[70] S. Khan, "Conflicts in betz limit and an alternative approach for wind turbines," in *Proc. IEEE Region 10 Symp. (TENSYMP)*, Jun. 2020, pp. 1438–1443.