Emulation of a 1.5MW Wind Turbine with a DC Motor

Trevor Hardy, Student Member, IEEE, and Ward Jewell, Fellow, IEEE,

Abstract—In order to provide a testing environment for generator control strategies, a wind turbine emulator was developed. First, a mathematical model was developed for the a 1.5MW wind turbine generator based on available modeling data. Using this data a function relating maximum power output and wind speed was developed along with a one-mass model of the turbine. The model was implemented in a software simulation along with controllers for a DC motor acting as the wind turbine and a dynamometer acting as the generator. The software simulation was implemented in LabVIEW and read in a data file containing high sample rate (20 Hz) wind speed data, calculated the optimal power from that wind speed, and generated control signals for the DC motor and dynamometer.

The effectiveness of the controller was found by calculating the mean-square error between the optimal power calculated by the simulation turbine model and the actual power produced by the DC motor/dynamometer combination. Two 800 second wind speed data files were used as test inputs for the emulator: a low-speed low-frequency wind profile and a turbulent or varying-speed wind profile. The results of the controller development using these two wind speeds show that the turbine inertia is a significant factor in following the optimal power output of a wind turbine. The time constant of the generator speed-tracking function specified in the modeling data also plays a significant role in how quickly the system can respond to changes in wind speed.

Index Terms—wind turbine, wind turbine emulation, DC motor control

I. INTRODUCTION

S the costs of fossil fuels, both financial and envi-A ronmental, become more obvious to society, there has been a renewed and increased effort in developing alternative energy sources, particularly as it relates to electrical power production. One of the leading prospects in this search for clean and commercially viable production are horizontalaxis wind turbines. The United States Department of Energy has headed up a collaboration of institutions to explore and define the effort of having 20% of the US energy produced by wind turbines by the year 2030 [1]. Wind turbines are currently being used in large-scale parallel configurations as alternatives to traditional fossil-fuel power plants and are being considered for local and neighborhood energy sources in smart- and micro-grid research [2]. In both uses, development of efficient and effective generator control strategies to allow for the maximum energy capture from the wind are of great importance.

Typically these strategies are tested using software models and simulations and, in academic settings, are rarely if ever able to be developed on real-world wind turbines. To provide a more realistic test environment for generator control techniques, an effort has been made to develop a wind-turbine

emulator where an electrical motor directly drives the shaft of the generator in a manner that mimics that of a turbine. The goal of this project was to develop such a test platform using as much real-world data as possible.

1

II. LITERATURE SURVEY

In building a wind-turbine emulator one of the key issues is the type of model to be used for software simulation. The input to the emulator is recorded wind speed data which is then applied to this turbine model to produce the reference power output which in turn is used in controlling the motor to produce the mechanical power for the emulator.

A fundamental aspect of turbine modeling is the aerodynamic effects, that is, a model which describes the affects of wind on the turbine blades. The fundamentals of this idealized model are widely available in literature ([3], [4], [5]) where the mechanical power output of the turbine is a function of parameters such as air density, turbine rotor area, turbine rotor speed, and wind speed. This relationship is fundamental to building a wind turbine emulator and in this case, a key modeling document was obtained which gave details for these parameters on a 1.5MW turbine [6].

Another key matter in the aerodynamic model is that of blade pitch control. Wind turbines intended to be used for commercial power production commonly have blades whose pitch can be varied as wind conditions change. To understand the role of blade pitch it is important to first define the regions of operation that a wind turbine typically experiences. If we imagine a steady breeze that starts at zero and steadily increases in speed, the first critical point the turbine encounters is the cut-in speed. Below this speed the wind turbine does not operate at all; once this speed is reached, the turbine begins to produce power. As the wind speed increases the next critical point that is reached is the rated wind speed. This is the point at which the generator is spinning as fast as it can and producing as much power as possible. Up to this point, the blades on the wind turbine have been pitched to capture as much energy from the wind as possible. Once the wind speed exceeds the rated wind speed, the turbine blades begin to adjust their pitch to shed or bypass some of the energy in the wind. The turbine continues to produce its maximum power even though the wind speed is larger than the turbines rated value. As the wind speed increases, the turbine continues to adjust the pitch of its blades until they reach their mechanical limit. At this point, the turbine must shut-down to prevent damage to itself. This point is called the cut-out speed [4] [6]. A graph showing these regions of operation is shown in figure 1 where θ represents the blade pitch angle.

978-1-4577-1002-5/11/\$26.00 ©2011 IEEE

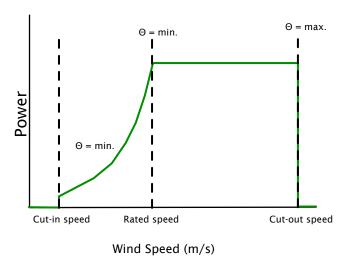


Fig. 1: Regions of wind turbine operation

The mechanical model of the turbine system as a whole (usually including the gearbox and appropriate shafts) are also important. There are existing models including higher order effects such as shaft elasticity and gearbox inertia ([7][8], [9], [10], [11]), often referred to as "two-mass" or "threemass models", depending on how many significant mechanical components are included in the model. It was determined early in development of the emulator that including these effects in our turbine model was beyond the scope of this project and thus they are neglected. A simple and significant model parameter that was included, though, was the inertia of the turbine. Mechanical inertia in wind turbine power generation plays a significant role by limiting the rate at which the system can respond to changing wind conditions, for better or for worse. There are many examples in literature on modeling the effects of inertia ([11], [12], [13]) and because of the obvious and significant effect it has on the frequency response of the system, it was included in this emulator as a component of the software simulation. This is referred to as a "one-mass" model.

Several publications were found where other groups had constructed wind turbine emulators using similar ideas as the ones proposed here. ([3], [5], [7], [8], [10], [14], [15], [16], [17]). To produce the mechanical power, the most common type of motor used was the DC motor due to the direct relationship between motor torque and rotor current. Some of these papers also outline control strategies for the motor, most involving a dual-loop control: an outer power control loop responds to changing wind conditions to produce an updated reference power and the inner control loop is a much faster loop designed to control the moment-by-moment fluctuations in the motor's torque. This inner loop used the reference power calculated by the outer loop to calculate a reference torque (given the system's rotational speed) which is directly proportional to rotor current. These efforts were considered when developing the control strategy for our wind turbine emulator.

III. MODEL DEVELOPMENT

A. Turbine Power Model Development

In designing the wind turbine emulator, a power curve for the turbine must first be developed. This curve defines the relationship between the maximum mechanical power the turbine can produce versus wind speed. In literature ([5]), the fundamental relationship is given as

$$P = \frac{1}{2}\rho A v^3 C_p \tag{1}$$

where ρ is the air density, A is the swept area of the turbine rotor, v is the wind speed, and C_p is the power coefficient of the turbine. The power coefficient is a measure of how efficiently the turbine is able to harvest energy from the wind and is itself a function of a parameter called the tip-speed ratio, λ .

 λ is the relationship between the speed at the tip of the turbine blade and the wind speed. λ is defined as

$$\lambda = \frac{\omega R}{v} \tag{2}$$

where ω is the mechanical rotational speed of the turbine shaft, R is the turbine radius, and v is the wind speed. Low values of λ indicate cases where the wind speed is relatively close to the rotational speed of the turbines. The shape of the typical power coefficient curve $(C_p \text{ vs } \lambda)$ is shown in Figure 2.

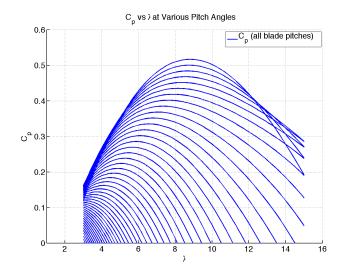


Fig. 2: Power Coefficient Family Of Curves

Modern high-power wind turbines have the ability to adjust the pitch of their blades, that is, the angle at which the edge of the blade meets the wind. Blade pitch alters the shape of the power coefficient curve, with higher blade pitches pushing the peak of the curve to lower values of λ but also lowering that peak overall. Including blade pitch as a factor in the power coefficient curve produces a family of curves as shown in Figure 2. Having this extra degree of freedom permits the turbine control system to operate more efficiently by allowing it to operate at the maximum point on a power coefficient curve. That is, for a given set of conditions, the blade pitch can be adjusted to find the point where C_p (and thus power output for those conditions) is maximized.

The wind turbine emulator being developed was based on a commercially available 1.5MW wind turbine. A modeling document ([6]) detailing many of the aspects of modeling this wind turbine and generator system was used as it included a function for computing the power coefficient curve for a particular turbine. Use of that function resulted in a family of curves shown in figure 3.

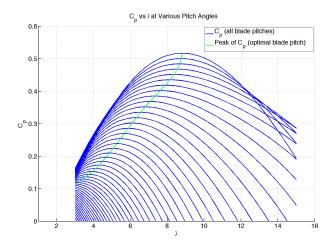


Fig. 3: Power Coefficient Family Of Curves With Peak \mathcal{C}_p Function

As can be seen, for a given turbine shaft speed ω and a given wind speed v, there is a specific blade pitch that produces the largest value of C_p . This represents the configuration that will capture the greatest amount of energy from the wind. Notice that either changes in turbine rotational speed or changes in wind speed (both effectively changing λ) can motivate a change in blade pitch to stay at the point of most efficient energy capture. Since the emulator is seeking to reproduce the effects of the mechanical turbine, including blade pitch, it was decided that blade pitch could be assumed to always be in the optimal position for the conditions at any given time and would be able to instantaneously adjust as the conditions changed. This simplified the software simulation by not requiring a blade pitch controller.

By assuming that λ can be maintained at the value necessary to produce a maximum value for C_p under a given set of conditions, it is possible to calculate the maximum power output from the turbine as a function of wind speed v and a function was fit that follows the peak point of each curve in the family. This function defines the maximum value of C_p as a function of λ which is itself a function of turbine shaft speed ω and wind speed v. Using equation 1, it is possible to form a separate family of curves where the turbine output power P is a function of wind speed and turbine shaft speed. This family of curves is shown in Figure 4 where each individual curve shows the output power for a given turbine shaft speed as a function of wind speed.

For a given wind speed, this family of curves shows that there are a variety of turbine shaft speeds that reside on the maximum C_p curve. For each wind speed, though, there is only one value of turbine shaft speed ω that produces the maximum

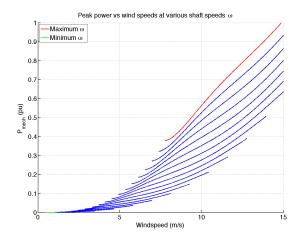


Fig. 4: Power Output vs Wind Speed Over Various Shaft Speeds

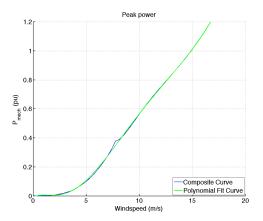


Fig. 5: Optimal Power Curve

power output. Looking at Figure 4 it is clear that at a given wind speed, the maximum power is generated on the highest curve available at that windspeed. Using this fact, Matlab was used to find the largest value of ω at each wind speed and a function was fit to these points; this function defines the maximum power curve for the wind turbine and is shown in Figure 5. Note that the maximum power output defines both a unique turbine blade pitch and turbine shaft speed.

Using this maximum power curves, it is possible to define a target power output for the turbine emulator based soley on windspeed.

B. Turbine Mechanical Model Development

Though essential, the mathematical model of the mechanical system can extend far beyond the maximum power output curve. Other factors that are commonly considered are turbine inertia, dynamic responses of the turbine/gearbox system (accounting for shaft elasticity, gear box inertia, etc), and tower shadow effects. For the purposes of this project, only the inertia of the turbine were included.

Turbine inertia is commonly expressed as the inertia time

constant H which is defined as

$$H = \frac{J\omega^2}{S} \tag{3}$$

where J is the rotational mechanical inertia of the turbine, ω is the mechanical rotational speed, and S is the apparent rated power of the system. These factors are used to calculate what is called the natural time constant of the turbine.

$$\tau_0 = \frac{1}{3}H\tag{4}$$

Based on the work of [13], the actual time constant of the system is also a function of wind speed.

$$\tau = \tau_0 \frac{v_{rated}}{v} \tag{5}$$

Using this function and the current wind speed, a low-pass filter with a dynamic cut-off frequency was used to simulate the effects of the turbine's inertia. The filter was applied to the recorded wind speed, removing the fast-changing, high-frequency components of the wind speed to which the turbine could not respond. It is this filtered wind speed that is used as the input to the optimal power curve developed in section III-A to determine the reference power that the emulator will attempt to track.

IV. EMULATOR DESIGN

A. Software Simulation

To provide effective emulation of the mechanical turbine characteristics, the previously discussed (section III) mechanical model components were implemented in software, specifically, LabVIEW. The goal of the software simulation is to calculate a target power output taking into account the recorded wind speed, the calculated maximum power curve, the turbine model, and any real-time values to complete these calculations. In this case, those necessary values were the DC motor's torque and speed. A general view of the system block diagram can be seen in Figure 6.

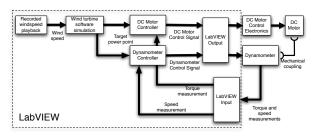


Fig. 6: Turbine block diagram of the entire wind emulation system

A diagram of the data-flow for the turbine simulation is shown in Figure 7. The simulation starts with reading in recorded wind speed data provided by the National Renewable Energy Lab from an observation tower of 120m in height. This data is used as the input to a look-up table which produces a target power point. To account for the turbine's inertia, a low-pass filter (LPF) is applied to this target power point, the cut-off frequency of which is determined using the time constant found for the turbine based on its inertia and the current wind

speed (see section III-B). This filtered power signal is the target power point that is then used as the reference to control the motor.

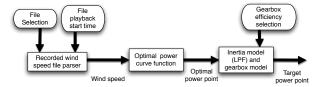


Fig. 7: Turbine model block diagram

B. DC Motor And Dynamometer Control

The calculated target power point (provided by the turbine simulation) is the central piece of data that is used by the DC motor and dynamometer controllers. In concert, these two pieces of equipment need to be controlled in such a way as to, as closely as possible, produce the target amount of power as the wind speed dictates. Obviously, for any given target power point there are a large number of torque and speed combinations that will yield a valid result. Based on current literature ([3], [8]), the common "division of labor" between the turbine control and generator control is to allow the generator to regulate the shaft speed and the turbine to regulate the torque. This technique was applied when developing the controllers for each unit.

The controllers were implemented using time-domain functions (rather than frequency domain transfer functions) due to their ease of implementation in LabVIEW. A basic PID controller was implemented and the specific gain values were found by trial and error during system development. The block diagram of the controller used is shown in Figure (8). The controller for the dynamometer is similar is structure, only controlling for speed rather than torque. There were two small additions made to the standard PID control structure: an addition of an input limiter to the integrator block to prevent integrator saturation and the second was a smoothing filter on the output of the derivative function. To account for the inertia effects of the turbine, a similar low-pass filter was placed on the output of the controller block to limit the rate at which the controller could request the system to respond.

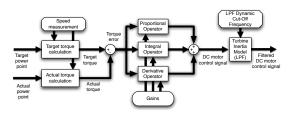


Fig. 8: DC Motor Controller Block Diagram

Note that the dynamometer controller and motor controller have no communication with each other inside the software simulation. Ideally, the dynamometer would be replaced with an actual generator which would have its own independent controller. The current emulator architecture affords this easily. The only information that is passed between the turbine and controller and dynamometer controller is the torque and speed which is "communicated" through the common mechanical linkage.

C. Mechanical System

The physical set-up for the emulator consists of a DC motor used to emulate the mechanical power output of the turbine, an insulated-gate bipolar transistor (IGBT) chopper/inverter (and its controller) which drives the DC motor, a dynamometer to emulate the mechanical loading that a generator would provide to the turbine, and high voltage (~200V) DC power supply. Both the DC motor and the dynamometer received a control signal from National Instruments analog input/output hardware that interfaced with the LabVIEW software simulation (see following section). The control signal for the dynamometer came from this I/O hardware and was directly applied to a control input on the dynamometer. The control signal was a varying voltage over a specified range that caused the dynamometer to produce a load that was proportional to that signal.

The electronics to control the DC motor were more complicated. A voltage from the I/O hardware (generated by the software simulation) was fed to a separate LabVolt controller. This controller accepts an analog input voltage and produces three control signals to be used on the three phase insulated-gate bipolar transistor (IGBT) chopper/inverter. The controller has a variety of modes that determine the form of those output voltages but for this purpose, we needed only a single-phase pulse-width modulated (PWM) signal. By applying a high input voltage (~200VDC) on the input of the IGBT chopper/inverter, the signals from the controller modulate this fixed higher voltage down to a lower voltage of an arbitrary value. This modulated DC voltage is then fed to the DC motor. The DC motor and the dynamometer were attached through a belt provided by LabVolt for this purpose.

A block diagram of the mechanical system is shown in Figure 9 and a photograph of the set-up is shown in Figure 10.

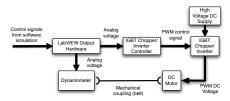


Fig. 9: Block Diagram of Mechanical System

V. EXPERIMENTS

The emulator was test in its ability to operate at the calculated target power point under a variety of wind conditions. The National Renewable Energy Lab (NREL) was able to provide three recorded wind speed files from a 120m tall wind measurement site in Colorado. Each file characterizes three separate wind conditions: high-speed winds, low-speed winds, and turbulent winds. These files were set-up to be readin by the emulator and used to calculate the target power point,



Fig. 10: Photograph of Mechanical System

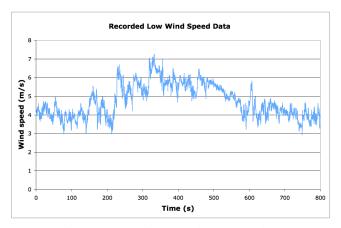


Fig. 11: Low Wind Speed Data vs. Time

which then drove the controllers' behavior. Plots of these wind profiles can be seen in Figures 11 and 12.

To measure the effectiveness of the emulator, an error signal was defined as the difference between the theoretical maximum power at a given wind-speed and the actual power output of the emulator at the moment. This error signal was then used to calculate the mean square error (MSE) on a rolling, real-time basis with the value at the conclusion of

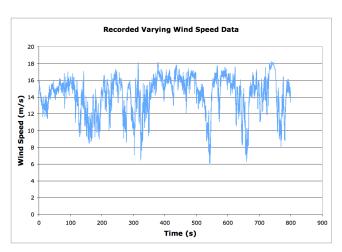


Fig. 12: Varying Wind Speed Data vs. Time

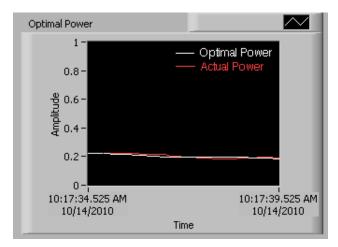


Fig. 13: Typical Response of System Under Nominal Controller Weights to Low-Speed Wind Profile

the run representing the total MSE for that run. The MSE is defined as

$$MSE = \frac{E^2}{n} \tag{6}$$

where E is the error signal defined above and n is the number of data points over which an error is calculated. MSE is then, as the name implies, a measurement of average error over a given dataset where larger deviation in error are weighted more heavily.

VI. RESULTS

Most of the effort during the experimentation was made in adjusting the weighting for the PID controllers for the DC motor (turbine emulation) and dynamometer (generator emulation). More emphasis was put into the DC motor controller as it is a permanent fixture of the emulator and the dynamometer is a stand-in until a fully-functional generator can be used. That being said, the turbine and the generator are mechanically coupled and both work together to produce optimal power generation; it is not possible to entirely neglect the dynamometer controller and expect the emulator as a whole to work.

Figures 13 and 14 show examples of how the system responded during typical portions of the low-speed and varying-speed runs under nominal but non-optimized controller weights.

Generally speaking, the MSE values for the low-speed run were very low and much of this can be attributed to the relatively slow changes in wind-speed for this data set. The varying-speed runs benefited from the wind-speed frequently crossing-into and maintaining a speed above the rated wind speed for a sustained period of time. When the wind-speed is above this threshold the power output of the system is constant at the maximum rating providing a fixed target that serves to damp out the system by saturating the outputs. The wind speed is above the rating value for approximately half of the time during the varying-speed runs.

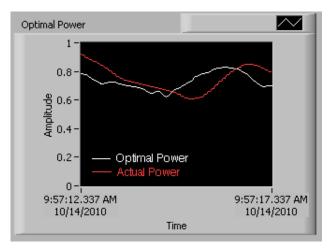


Fig. 14: Typical Response of System Under Nominal Controller Weights to Varying-Speed Wind Profile

It was also noticed during the course of multi-hour experimental sessions that there appeared to be parameter drift in the system as it warmed up. No attempt was made to quantify the severity of the parameter drift from a "cold" to "warmed-up" system.

VII. CONCLUSION

Based on the results of the development and experimentation thus far, the emulator in its current state provides reasonable emulation of a wind turbine power output and can be used for development of generator control strategies. The system has been demonstrated to track the changes in power based on the wind speed and shows the inertial effects of the turbine which limit the system's ability to respond quickly to changes in wind speed.

It was discovered early in the process of finding good weights for the controllers that a typical failing of a classic PID controller was a significant issue: integrator saturation and/or wind-up. Since the integrator function of the PID is constantly accumulating the system error in an effort to force the controller to respond to small errors that persist over time, it acts as if it has a long memory. This can cause problems as a large but temporary error is not forgotten or neglected by the integrator and causes its output to skew heavily in one direction for a long period of time, even past the point where such a skew is beneficial. Similarly, if conditions exist where a small error persists over a long period of time (such as an actuator at its mechanical limit or other control weights dominating the response of the system), the integrator will accumulate this error and its output will attempt to correct all of this error even if the conditions that originated in the error have passed. If it is not possible to keep the error signal into the integrator at a low value, the integrator can easily come to have a significant negative impact on the overall control effectiveness.

There are known strategies for dealing with integrator saturation and wind-up but most of them require a more complex analysis of the system and thus were beyond the scope of this project. One strategy, though, was employed with some

success and that was limiting the input of the error signal into the integrator to a very small range, say 10% of its actual value. This prevents the integrator from being saturated by large transient errors but doesn't address the issue of wind-up.

In addition to the integrator input limitation, the derivative block also was slightly modified from the strict PID model to help accomplish better control. The derivative block uses a point-by-point derivative using the two latest input points and the time between them to find the necessary slope. For reasons that are not clear, there are times, though, that the input signal does not change from point-to-point even though the signal as a whole is experiencing rapid change. This would lead to alternating derivative calculations oscillating between zero and a relatively high value. To clean rapid oscillation, a first-order smoothing filter was added on the output of the derivative block.

VIII. FUTURE WORK

The turbine emulator designed and implemented here was able to accomplish basic turbine emulation in a sufficient manner though there is much room for improvement.

Before looking at improvements in specific areas of the emulator, there is one significant next step that would legitimize this emulator as a useful development tool: validating the emulator against real-world data from from an operating wind turbine. If a set of power output and wind speed versus time data could be obtained for the turbine being emulated here it would prove invaluable in determining how valid this existing emulator is and what efforts could be made to develop it to the point of being a good representation of the real system. If a validated emulator could be implemented, it would allow for significant research opportunities in many areas including optimal power point tracking and generator speed selection during fluctuating wind speeds.

A. Programming/Simulation

The DC motor controller was implemented with time domain functions due to ease of implementation and possible hardware limitations. Moving both the turbine model and the controller to frequency domain functions would provide greater flexibility as most controls engineers are more comfortable working with frequency domain transfer functions. The feasibility of this change may hinge on the acquisition of new LabVIEW hardware to support the real-time environment. Additional expertise in LabVIEW programming would also be an asset in implementing the frequency-domain conversion and/or real-time environment.

Also due to limited expertise in LabView programming, the generator speed-tracking filter was designed to be set at 60 s but currently the filter appears to effectively operate at 10 s. This difference in filter cut-off frequency is significant and will have a significant impact on the systems response to changing wind conditions and will necessitate the use of new controller weights.

Integrator wind-up and/or saturation resulted in a lower performing controller as compared to those where the integrator weight was set to zero. An intelligent and effective means to counter-act these common problems with the integrator could result in improved control during periods of semi-steady-state operation (close to constant wind-speed).

B. Modeling

Integration of the effects of a pitch controller on power output when the wind speed is greater than the rated wind speed would provide more realistic conditions for the emulator, particularly for high wind speed situations. As the emulator functions right now these conditions produce a static state at maximum power.

The mechanical model for the turbine at present is limited and offers much room for improvement. There is a great deal of literature that provide more realistic mechanical models including items such as shaft/gear-box elasticity and gear-box moment of inertia. The effect of these additional mechanical parameters would provide greater model accuracy during transient conditions such as large changes in wind-speed or fault-conditions.

REFERENCES

- [1] U. S. D. of Energy. Office of Energy Efficiency, R. Energy, and N. R. E. L. (U.S.), Wind Power in America's Future: 20 % Wind Energy by 2030. Mineola, New York: Dover Publications, Jan 2010.
- [2] W.-T. Huang and W.-C. Yang;, "System steady-state analysis of a low-voltage microgrid with various distributed energy resources," *IEEE Conference on Cybernetics and Intelligent Systems (CIS)*, 2010, pp. 237 242, 2010.
- [3] K. Han and G. zhu Chen, "A novel control strategy of wind turbine mppt implementation for direct-drive pmsg wind generation imitation platform," *IEEE 6th International Power Electronics and Motion Control Conference*, 2009, pp. 2255–2259, 2009.
- [4] L. Pao and K. Johnson, "A tutorial on the dynamics and control of wind turbines and wind farms," *American Control Conference*, 2009, pp. 2076–2089, 2009.
- [5] W. Li, D. Xu, W. Zhang, and H. Ma, "Research on wind turbine emulation based on dc motor," *IEEE Conference on Industrial Electronics* and Applications, 2007., pp. 2589–2593, 2007.
- [6] K. Clark, N. W. Miller, and J. J. Sanchez-Gasca, "Modeling of ge wind turbine-generators for grid studies," GE Energy, Tech. Rep., Sep 2009.
- [7] L. Lopes, J. Lhuilier, A. mukherjee, and M. Khokhar, "A wind turbine emulator that represents the dynamics of the wind turbine rotor and drive train," *IEEE 36th Power Electronics Specialists Conference*, 2005, pp. 2092–2097, 2005.
- [8] M. Chinchilla, S. Arnaltes, and J. Rodriguez-Amenedo, "Laboratory set-up for wind turbine emulation," *IEEE International Conference* on Industrial Technology, 2004., vol. 1, pp. 553–557, 2004.
- [9] G. Raina, O. Malik, and A. Kumar, "Wind oscillatory and gusting torque effect on a power system," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-21, no. 6, pp. 767–776, 1985.
- [10] B. Rabelo, W. Hofmann, and M. Gluck, "Emulation of the static and dynamic behaviour of a wind-turbine with a dc-machine drive," *IEEE* 35th Annual Power Electronics Specialists Conference, 2004., vol. 3, pp. 2107–2112 Vol.3, 2004.
- [11] J. Slootweg, S. D. Haan, H. Polinder, and W. Kling, "General model for representing variable-speed wind turbines in power system dynamics simulations," *IEEE Power Engineering Review*, vol. 22, no. 11, pp. 56– 56, 2002.
- [12] J. Morren, J. Pierik, and S. D. Haan, "Inertial response of variable speed wind turbines," *Electric Power Systems Research*, vol. 76, no. 11, pp. 980–987, 2006.
- [13] C. Tang, M. Pathmanathan, W. Soong, and N. Ertugrul, "Effects of inertia on dynamic performance of wind turbines," *Australasian Universities Power Engineering Conference*, 2008., pp. 1–6, 2008.
- [14] M. Arifujjaman, M. Iqbal, and J. Quaicoe, "Maximum power extraction from a small wind turbine emulator using a dc - dc converter controlled by a microcontroller," *International Conference on Electrical and Com*puter Engineering, 2006., pp. 213–216, 2006.

- [15] H. Kojabadi, L. Chang, and T. Boutot, "Development of a novel wind turbine simulator for wind energy conversion systems using an invertercontrolled induction motor," *IEEE Transactions on Energy Conversion*, vol. 19, no. 3, pp. 547–552, 2004.
- [16] B. Gong and D. Xu, "Real time wind turbine simulator for wind energy conversion system," *IEEE Power Electronics Specialists Conference*, 2008., pp. 1110–1114, 2008.
- [17] W. Sikwete and P. Barendse, "Wind turbine emulator," Master's thesis, University of Capetown, Jan 2008.



Trevor Hardy (S '08) received the Bachelors degree in Electrical Engineering from LeTourneau University, Longview, Texas, in 2001, and the M.S. degree in Electrical Engineering from Wichita State University, Wichita, KS, in 2010. Currently, he is pursuing Ph.D. degree and working as a Graduate Research Assistant at Wichita State University.



Ward Jewell (M'77 - F'03) received the B.S.E.E. degree from Oklahoma State University, Stillwater, in 1979, the M.S.E.E. degree from Michigan State University, East Lansing, in 1980, and the Ph.D. degree from Oklahoma State University, in 1986. He has been with Wichita State University, Wichita, KS, since 1987, where he is currently a Professor of Electrical Engineering. He is the Site Director at the Power System Engineering Research Center (PSerc), Wichita State University. His current research interests include electric power quality and

advanced energy technologies.