ISSN:2229-6107

INTERNATIONAL JOURNAL OF PURE AND APPLIED SCIENCE & TECHNOLOGY

E-mail :
editor.ijpast@gmail.com
editor@ijpast.in

www.ijpast.in

ISSN 2229-6107 www.ijpast.in

Vol 13,Issuse 2.May 2023

DESIGN FOR VARIABLE SPEED WIND TURBINE ENERGY SYSTEM USING PSO

[1]DR.K. CHANDRAM, [2]D. KOUSALYA,[3]G. SAI SPANDANA,[4]M. PAVITRA

Abstract:

Renewable energy from wind is the safest form of energy. Wind turbine based energy generators have the potential to generate high amount of electric power if there is a proper wind velocity and control mechanisms. This can certainly reduce the dependency on solar photovoltaic based energy systems, which needs huge space to install the solar photovoltaic panels. However, the output power of wind turbine is affected by the uncertain wind velocity. The output mechanical power has to be properly controlled. Hence, the wind energy system efficacy depends on how well this uncertainty is addressed. The major challenge is to design and control the wind turbine systems that has a suitable mediator between the power generator and the load, which counters the damage to the load due to variable voltages produced by the varying wind velocity. Keeping this in view, this paper implements all-important PID control design methods for wind energy application and recommends the most suitable method for its controller design. The overall analysis is presented via detailed quantitative results that are evaluated with the help of time-domain performance index parameters

Introduction

The overwhelming reliance of the global economy on fossil fuels and environmental concerns have prompted a shift in attention toward nonconventional means of producing power. Wind power is the most rapidly expanding renewable energy source in this context of increasing energy market diversity [1]. For a long time, the most common kind of wind turbine was one with a simple control system designed to save operating expenses and upkeep [1]. Electronic converters and mechanical actuators have become more popular as a result of the growing size of turbines and the rising penetration of wind energy into the utility networks of leading nations. In order to actively regulate the absorbed energy, these devices integrate additional design degrees of freedom. As an interface to the power grid, static converters allow for variable-speed operation up to the rated speed. Variable speed control seems to be a viable alternative for improving the functioning of wind turbines in the face of environmental disturbances including random wind variations, wind shear, and tower shadows [2]. From a control system perspective, wind energy conversion systems provide unique difficulties. Due to their nonlinear nature and susceptibility to significant cyclic disturbances, wind turbines may experience excitation of the weakly damped vibration modes of the drive-train and tower (see [1,3]). Furthermore, because to the unique working circumstances, it is challenging to construct mathematical models that effectively represent the dynamic behaviour of wind turbines.

[1]PROFESSOR, [2,3&4] UG SCHOLAR

DEPARTMENT OF EEE,

MALLA REDDY ENGINEERING COLLEGE FOR WOMEN, HYDERABAD

The present trend toward bigger and more adaptable wind turbines makes this work much more complex. Robust control solutions may help mitigate the effects of inaccurate models by guaranteeing stability and specific performance characteristics in spite of model errors. Turbines with variable speed and pitch increase the complexity of the control difficulties, as shown in [4-6]. Multiple controls are required for optimal performance of this kind of turbine (see to [7,8] for details). In this study, we suggest a novel approach to controlling horizontal-axis wind turbines (HAWTs) that may vary both in speed and pitch. This regulation is accomplished by a proportional integral (PI) controller for the blade pitch angle and a nonlinear dynamic chattering torque control approach. The new control mechanism permits a fast change in the amount of electricity produced by the wind turbines. This suggests that WT power generation may be adjusted up or down depending on network demand for electricity. All other state variables, such as the rotational speeds of the turbines and generators, as well as the smooth and appropriate development of the control variables, guarantee this electrical power tracking.

Modelling of Systems

A rotor assembly, transmission, and generator make up the wind turbine. The kinetic energy of the wind is converted into mechanical power by the rotor of the wind turbine. Condensed version of thein [12-14], a rotor was used. The following equation describes the relationship between wind speed and the resulting mechanical power in this model.

$$P_m(u) = \frac{1}{2}C_p(\lambda,\beta)\rho\pi R^2u^3$$

where ρ is the air density, R is the radius of the rotor, u is the wind speed, Cp is the power coefficient of the wind turbine, β is the pitch angle, and λ is the tip-speed ratio given by

$$\lambda = \frac{R\omega_r}{u}$$

where ωr is the rotor speed. Thus, changes in the wind speed or rotor speed produce changes in the tip-speed ratio, leading to power coefficient variation; thus, the generated power is affected. The aerodynamic torque coefficient is related to the power coefficient as follows,

$$P_m = \omega_r T_a$$

the aerodynamic torque expression is described as

$$T_a = \frac{1}{2}C_q(\lambda,\beta)\rho\pi R^3u^2$$

where

$$C_q(\lambda,\beta) = \frac{C_p(\lambda,\beta)}{\lambda}$$

For a perfectly rigid low-speed shaft, a single-mass model for a wind turbine can be considered

$$J_t\dot{\omega}_r = T_a - K_t\omega_r - T_g$$

where Jt is the turbine total inertia (kg m2), Kt is the turbine total external damping (Nm rad−1 s), Ta is the aerodynamic torque (Nm), and Tg is the generator torque (Nm). The scheme of the one-mass model is provided in Figure 1.

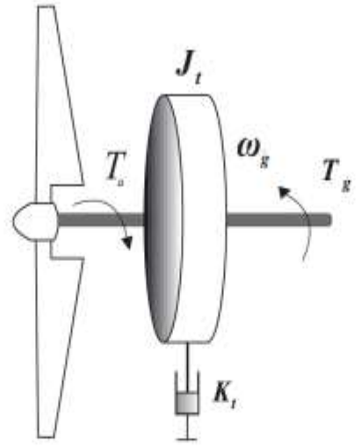

Figure 1. One-mass model of a wind turbine.

The Short Synopsis of a Simulator (FAST)

When it comes to calculating the severe and fatigue loads of two- and three-bladed HAWTs, the FAST algorithm [9] is the gold standard aeroelastic simulator. Because Germanischer Lloyd Windemere examined this simulator in 2005 and considered it appropriate for the simulation of onshore wind turbine loads for design and certification [18], it was selected for validation. The complex turbine controls may be implemented in Simulink's user-friendly block diagram format thanks to the MATLABR interface built between FAST and Simulink. To include the FAST motion equations (in an S-function), the FAST subroutines are coupled using a MATLAB standard gateway subroutine. This provides a huge amount of leeway for adjusting the simulated controls of a wind turbine. The full nonlinear aeroelastic wind turbine equations of motion are accessible in FAST, allowing for their use in the design and simulation of control modules for the generator torque, nacelle yaw, and pitch in the Simulink environment. Blocks that integrate accelerations in different degrees of freedom to get velocities and displacements are included in the wind turbine block, along with an S-function block containing the FAST motion equations. This necessitates writing the equations of motion in the FAST S-function and then using one of Simulink's solvers to get the answers.

PSO

In computational science, particle swarm optimization (PSO)[1] is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. It solves a problem by having a population of candidate solutions, here dubbed particles, and moving these particles around in the search-space according to simple mathematical formula over the particle's position and velocity. Each particle's movement is influenced by its local best known position, but is also guided toward the best known positions in the search-space, which are updated as better positions are found by other particles. This is expected to move the swarm toward the best solutions. PSO is originally attributed to Kennedy, Eberhart and Shi[2][3] and was first intended for simulating social behaviour,[4] as a stylized representation of the movement of organisms in a bird flock or fish school. The algorithm was simplified and it was observed to be performing optimization. The book by Kennedy and Eberhart[5] describes many philosophical aspects of PSO and swarm intelligence. An extensive survey of PSO applications is made by Poli.[6][7] Recently, a comprehensive review on theoretical and experimental works on PSO has been published by Bonyadi and Michalewicz.[1]

PSO is a metaheuristic as it makes few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. Also, PSO does not use the gradient of the problem being optimized, which means PSO does not require that the optimization problem be differentiable as is required by classic optimization methods such as gradient descent and quasi-newton methods. However, metaheuristics such as PSO do not guarantee an optimal solution is ever found

Neighbourhoods and topologies

The topology of the swarm defines the subset of particles with which each particle can exchange information.[28] The basic version of the algorithm uses the global topology as the swarm communication structure.[10] This topology allows all particles to communicate with all the other particles, thus the whole swarm share the same best position g from a single particle. However, this approach might lead the swarm to be trapped into a local minimum,[29] thus different topologies have been used to control the flow of information among particles. For instance, in local topologies, particles only share information with a subset of particles.[10] This subset can be a geometrical one[30] – for example "the m nearest particles" – or, more often, a social one, i.e. a set of particles that is not depending on any distance. In such cases, the PSO variant is said to be local best (vs global best for the basic PSO).

A commonly used swarm topology is the ring, in which each particle has just two neighbours, but there are many others.[10] The topology is not necessarily static. In fact, since the topology is related to the diversity of communication of the particles,[31] some efforts have been done to create adaptive topologies (SPSO,[32] APSO,[33] stochastic star,[34] TRIBES,[35] Cyber Swarm,[36] and C-PSO[37])

Inner workings

There are several schools of thought as to why and how the PSO algorithm can perform optimization.

A common belief amongst researchers is that the swarm behaviour varies between exploratory behaviour, that is, searching a broader region of the search-space, and exploitative behaviour, that is, a locally oriented search so as to get closer to a (possibly local) optimum. This school of thought has been prevalent since the inception of PSO.[3][4][12][16] This school of thought contends that the PSO algorithm and its parameters must be chosen so as to properly balance between exploration and exploitation to avoid premature convergence to a local optimum yet still ensure a good rate of convergence to the optimum. This belief is the precursor of many PSO variants, see below.

Another school of thought is that the behaviour of a PSO swarm is not well understood in terms of how it affects actual optimization performance, especially for higher-dimensional search-spaces and optimization problems that may be discontinuous, noisy, and time-varying. This school of thought merely tries to find PSO algorithms and parameters that cause good performance regardless of how the swarm behaviour can be interpreted in relation to e.g. exploration and exploitation. Such studies have led to the simplification of the PSO algorithm, see below.

Convergence

In relation to PSO the word convergence typically refers to two different definitions:

Convergence of the sequence of solutions (aka, stability analysis, converging) in which all particles have converged to a point in the search-space, which may or may not be the optimum,

Convergence to a local optimum where all personal bests p or, alternatively, the swarm's best known position g, approaches a local optimum of the problem, regardless of how the swarm behaves.

Convergence of the sequence of solutions has been investigated for PSO.[15][16][17] These analyses have resulted in guidelines for selecting PSO parameters that are believed to cause convergence to a point and prevent divergence of the swarm's particles (particles do not move unboundedly and will converge to somewhere). However, the analyses were criticized by Pedersen[22] for being oversimplified as they assume the swarm has only one particle, that it does not use stochastic variables and that the points of attraction, that is, the particle's best known position p and the swarm's best known position g, remain constant throughout the optimization process. However, it was shown[38] that these simplifications do not affect the boundaries found by these studies for parameter where the swarm is convergent. Considerable effort has been made in recent years to weaken the modelling assumption utilized during the stability analysis of PSO,[39] with the most recent generalized result applying to numerous PSO variants and utilized what was shown to be the minimal necessary modeling assumptions.[40]

Convergence to a local optimum has been analyzed for PSO in[41] and.[42] It has been proven that PSO needs some modification to guarantee finding a local optimum.

This means that determining convergence capabilities of different PSO algorithms and parameters still depends on empirical results. One attempt at addressing this issue is the development of an "orthogonal learning" strategy for an improved use of the information already existing in the relationship between p and g, so as to form a leading converging exemplar and to be effective with any PSO topology. The aims are to improve the performance of PSO overall, including faster global convergence, higher solution quality, and stronger robustness.[43] However, such studies do not provide theoretical evidence to actually prove their claims.

Adaptive mechanisms

Without the need for a trade-off between convergence ('exploitation') and divergence ('exploration'), an adaptive mechanism can be introduced. Adaptive particle swarm optimization (APSO) [44] features better search efficiency than standard PSO. APSO can perform global search over the entire search space with a higher convergence speed. It enables automatic control of the inertia weight, acceleration coefficients, and other algorithmic parameters at the run time, thereby improving the search effectiveness and efficiency at the same time. Also, APSO can act on the globally best particle to jump out of the likely local optima. However, APSO will introduce new algorithm parameters, it does not introduce additional design or implementation complexity nonetheless.

Variants

Numerous variants of even a basic PSO algorithm are possible. For example, there are different ways to initialize the particles and velocities (e.g. start with zero velocities instead), how to dampen the velocity, only update pi and g after the entire swarm has been updated, etc. Some of these choices and their possible performance impact have been discussed in the literature.[14]

A series of standard implementations have been created by leading researchers, "intended for use both as a baseline for performance testing of improvements to the technique, as well as to represent PSO to the wider optimization community. Having a well-known, strictly-defined standard algorithm provides a valuable point of comparison which can be used throughout the field of research to better test new advances."[10] The latest is Standard PSO 2011 (SPSO-2011).[45]

Hybridization

New and more sophisticated PSO variants are also continually being introduced in an attempt to improve optimization performance. There are certain trends in that research; one is to make a hybrid optimization method using PSO combined with other optimizers,[46][47][48] e.g., combined PSO with biogeography-based optimization,[49] and the incorporation of an effective learning method.[43]

Alleviate premature convergence

Another research trend is to try and alleviate premature convergence (that is, optimization stagnation), e.g. by reversing or perturbing the movement of the PSO particles,[19][50][51][52] another approach to deal with premature convergence is the use of multiple swarms[53] (multi-swarm optimization). The multi-swarm approach can also be used to implement multi-objective optimization.[54] Finally, there are developments in adapting the behavioural parameters of PSO during optimization.[44][24]

Simplifications

Another school of thought is that PSO should be simplified as much as possible without impairing its performance; a general concept often referred to as Occam's razor. Simplifying PSO was originally suggested by Kennedy[4] and has been studied more extensively,[18][21][22][55] where it appeared that optimization performance was improved, and the parameters were easier to tune and they performed more consistently across different optimization problems.

Another argument in favour of simplifying PSO is that metaheuristics can only have their efficacy demonstrated empirically by doing computational experiments on a finite number of optimization problems. This means a metaheuristic such as PSO cannot be proven correct and this increases the risk of making errors in its description and implementation. A good example of this[56] presented a promising variant of a genetic algorithm (another popular metaheuristic) but it was later found to be defective as it was strongly biased in its optimization search towards similar values for different dimensions in the search space, which happened to be the optimum of the benchmark problems considered. This bias was because of a programming error, and has now been fixed.[57] Initialization of velocities may require extra inputs. The Bare Bones PSO variant[58] has been proposed in 2003 by James Kennedy, and does not need to use velocity at all. Another simpler variant is the accelerated particle swarm optimization (APSO),[59] which also does not need to use velocity and can speed up the convergence in many applications. A simple demo code of APSO is available.[60]

Multi-objective optimization

PSO has also been applied to multi-objective problems,[61][62][63] in which the objective function comparison takes Pareto dominance into account when moving the PSO particles and non-dominated solutions are stored so as to approximate the pareto front.

Binary, discrete, and combinatorial

As the PSO equations given above work on real numbers, a commonly used method to solve discrete problems is to map the discrete search space to a continuous domain, to apply a classical PSO, and then to demap the result. Such a mapping can be very simple (for example by just using rounded values) or more sophisticated.[64]

However, it can be noted that the equations of movement make use of operators that perform four actions:computing the difference of two positions. The result is a velocity (more precisely a displacement)multiplying a velocity by a numerical coefficient

adding two velocities applying a velocity to a position

Usually a position and a velocity are represented by n real numbers, and these operators are simply -, *, +, and again +. But all these mathematical objects can be defined in a completely different way, in order to cope with binary problems (or more generally discrete ones), or even combinatorial ones.[65][66][67][68] One approach is to redefine the operators based on sets

Method of Regulation

The requested torque and pitch controls have been implemented in the straightforward block diagram format of Simulink, thanks to the MATLABR interface we built between FAST and Simulink. Figure 2 shows the open-loop FAST simulink model.

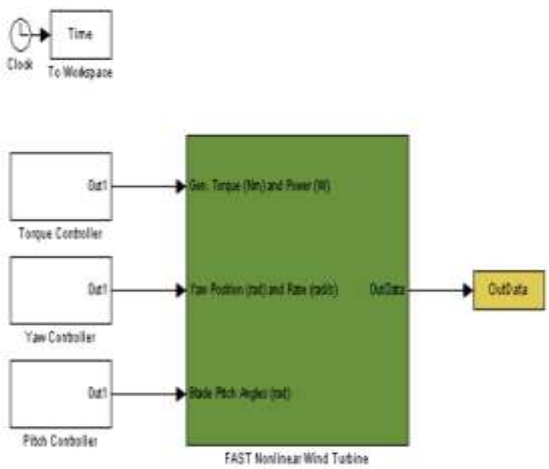

Figure 2. Simulink open-loop model.

The next sections present the proposed nonlinear dynamic torque and linear pitch controller designs.

Calculation Outcomes

FAST on MATLAB-Simulink was used to validate the NREL WP 1.5 MW wind turbine numerically. Table 1 summarizes the main features of wind turbines.

Table 1. Wind Turbine Characteristics.

Number of blades	3
Height of tower	82.39 m
Rotor diameter	70 m
Rated power	1.5 MW
Gearbox ratio	87.965
Nominal rotor speed (ω_n)	20 rpm

The wind inflow for the simulations is shown in Figure 3. A variable reference set point is imposed on the WT electrical power. When the wind park manager requires a given electrical power, he/she must dispatch this reference over different wind turbines and impose a variable reference for each turbine to meet a specific request for the grid. This wind inflow, for the simulated NREL WP 1.5-MW wind turbine, reaches wind speeds that are above the rated power operating conditions. From Figure 3, the rated windspeed for the wind turbine is 11.8 m/s, which coincides with the mean wind speed profile. Figure 3 also shows the reference power (right y-axes).

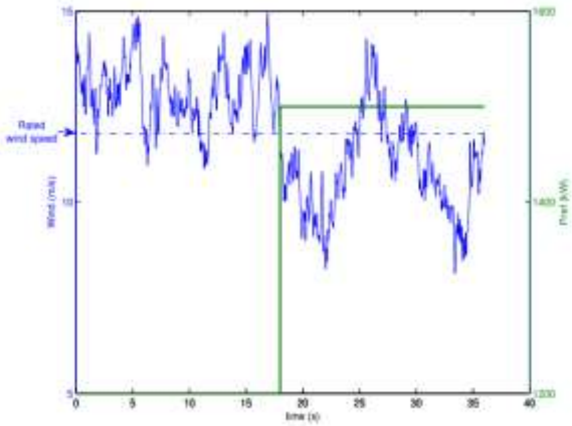

Figure 3. Wind speed profile with a mean of 11.8 m/s that corresponds to the rated wind speed of the WT (left y-axes). Reference power (right y-axes).

Managed Torque and Pitch With a = 1, Kp = 1, Ki = 1, and two values for K (different settling times), the FAST simulator calculates the torque and pitch control outputs. Specifically, these findings are contrasted with those obtained using the controllers suggested in [10] (Bukhezzar's controller) and [11] (Jonkman's controller). As can be seen in Figure 4, the rotor speed is very close to the nominal value (20 rpm) for all of the tested controllers as a result of the pitch control operation. As can be seen in Figure 5, when the reference electrical power is altered, an exponential convergence is seen with the Boukhezzar controller, and the target value is attained in about 5 seconds. However, as will be demonstrated below, the Jonkman controller achieves almost flawless power regulation at the expense of producing excessive loads that may easily outstrip the design load. Our suggested controller exhibits characteristics between those of Jonkman's and Boukhezzar's. When the parameter K = 1.5 106, the electrical power tracks the reference with a settling time of one second, as predicted [see Equation (10)]. Using K = 1.5 105 yields the same results, although with a lengthened settling period. To get a controller that is more in line with Jonkman's or Boukhezzar's, ours permits the settling time to be chosen. Our controller has limited convergence; thus it takes longer time than the Boukhezzar's controller to get to the reference power. Figure 5 (in its enlarged form) shows this pattern.

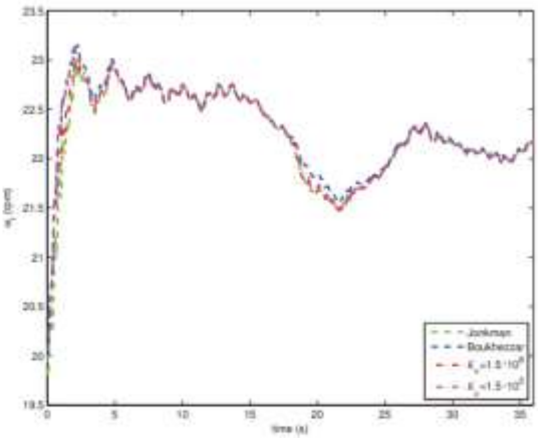

Figure 4. Rotor speed.

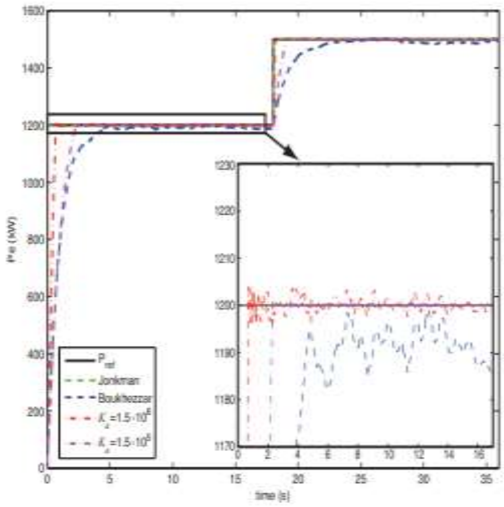

Figure 5. Power output.

Typical maximum pitch rates range from 18◦ /s for 600 kW research turbines to 8 ◦ /s for 5 MW turbines [23]. From Figure 6, for all the tested controllers, the blade pitch angle is always within the authorized variation domain without exceeding a variation of 10◦ /s.

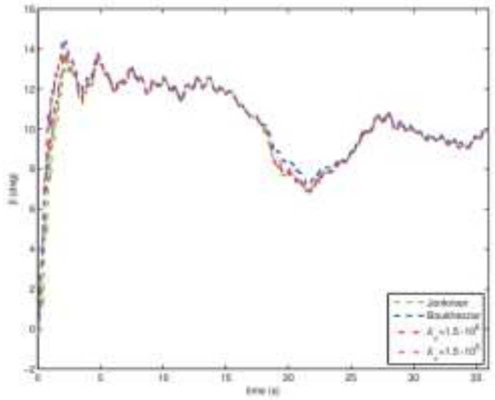

Figure 6. Pitch control.

Similar to the results achieved by the Jonkman and Boukhezzar controllers, the torque action of the suggested controller is smooth and suitable values are attained. Depending on the load, the generator may not be able to provide the required amount of electro-mechanical torque.

Conclusions

This research presents a WT controller optimized for use in high-turbulence wind environments. Strong performances in rotor speed and electrical power management are achieved with satisfactory control activity using the suggested controller. These findings demonstrate that the suggested controller enables a range of setpoints for the power provided by a WT. This success implies that WT power production can be scaled up or down in response to the network's power consumption and that WTs can take part in primary grid frequency control, allowing for a greater percentage of wind to be incorporated into electric networks without compromising on the quality of the generated electric power. Finally, we detail how the suggested controller is superior than the previously tested methods. • The suggested controller guarantees stability over limited intervals of time. Therefore, in comparison to exponentially stable controllers like [10], the suggested controller becomes closer to the target power reference.Settling time may be adjusted in the proposed controller by adjusting the values of a and K in Equation (4). Our controller's tuning allows for the production of intermediate controllers with settling times between those of the Jonkman and Boukhezzar controllers.

References

[1]. Burton, T.; Sharpe, D.; Jenkins, N.; Bossanyi, E. Wind Energy Handbook; Wiley: Chichester, UK, 2001.

[2]. Zinger, D.; Muljadi, E. Annualized wind energy improvement using variable speeds. IEEE Trans. Ind. Appl. 1997, 33, 1444–1447.

[3]. Kusiak, A.; Zhang, Z. Control of wind turbine power and vibration with a data-driven approach. Renew. Energy 2012, 43, 73–82.

[4]. Hassan, H.M.; Eishafei, A.L.; Farag, W.A.; Saad, M.S. A robust LMI-based pitch controller for large wind turbines. Renew. Energy 2012, 44, 63–71.

[5]. Sandquist, F.; Moe, G.; Anaya-Lara, O. Individual pitch control of horizontal axis wind turbines. J. Offshore Mech. Arctic Eng.-Trans. ASME 2012, 134, doi:10.1115/1.4005376.

[6]. Joo, Y.; Back, J. Power regulation of variable speed wind turbines using pitch control based on disturbance observer. J. Electr. Eng. Technol. 2012, 7, 273–280.

[7]. Diaz de Corcuera, A.; Pujana-Arrese, A.; Ezquerra, J.M.; Segurola, E.; Landaluze, J. H-infinity based control for load mitigation in wind turbines. Energies 2012, 5, 938–967.

[8]. Soliman, M.; Malik, O.P.; Westwick, D.T. Multiple Model MIMO Predictive Control for Variable Speed Variable Pitch Wind Turbines. In Proceedings of the American Control Conference, Baltimore, MD, USA, 30 June–2 July 2010.

[9]. Jonkman, J. NWTC Design Codes (FAST). Available online: http://wind.nrel.gov/designcodes/ simulators/fast/ (accessed on 8 March 2012).

[10]. Boukhezzar, B.; Lupu, L.; Siguerdidjane, H.; Hand, M. Multivariable control strategy for variable speed, variable pitch wind turbines. Renew. Energy 2007, 32, 1273–1287