MAXIMIZATION OF POWER CAPTURE IN WIND TURBINES USING ROBUST ESTIMATION AND LYAPUNOV EXTREMUM SEEKING CONTROL

by

TONY HAWKINS

B.S., Kansas State University, 2007

A THESIS

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Mechanical Engineering
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2010

Approved by:

Co-Major Professor
Dr. Warren N. White

Approved by:

Co-Major Professor
Dr. Guoqiang Hu

Abstract

In recent years, the concern has risen to establish clean sources for electric power generation. In 2009, Kansas established an RPS (Renewable Portfolio Standard) mandating utilities acquire 20% of their electricity from renewable energy by 2020 [32]. One of the most prominent renewable energy sources is wind energy. Utility companies now are investing more in wind capture systems to comply with this mandate. This increase in the manufacture of wind turbines has caused researchers to investigate methods to improve the efficiency of captured wind energy and where improvements can be made. This thesis takes a control theory approach to maximizing the power capture of a wind turbine using the concepts of robust estimation, nonlinear control, and Lyapunov-based maximization.

A two step control approach to optimize the power capture of a wind turbine is proposed. First, a robust controller is used to estimate unknown aerodynamic properties and regulate the wind turbine tip-speed ratio as it tracks a desired trajectory. Once the tip-speed ratio is regulated within a given tolerance, a Lyapunov-based control approach is developed to provide the robust controller with a desired trajectory to track. This is done by estimating the unknown coefficient of performance of the wind turbine. A discrete update law is then developed to alter the tip-speed ratio and the blade pitch of the wind turbine so that the coefficient of performance is maximized.

A simulation is provided of this control strategy and tested under time varying wind conditions and measurement noise in order to demonstrate the controller's performance. The system simulated is intended to emulate a commercial wind turbine operating in a realistic environment. A detailed discussion of the simulation model, control scheme, and results will be provided to supplement the theoretical controller development, as well as future work for this control application.

Table of Contents

List of Figures

List of Tables

Acknowledgements

The research for this thesis was supported by the Kansas State University Electrical Power Affiliates Program (EPAP) and the support is greatly appreciated. I would also like to acknowledge Dr. Kathryn Johnson, Colorado School of Mines, and the National Renewable Energy Laboratory for their assistance in providing data and software used for the wind turbine coefficient of performance modeling.

Many thanks to my major professors Dr. Warren White and Dr. Guoqiang Hu. I learned so much from both of them and would like to personally thank them for all of their time that they spent working with me on this research. I would also like to acknowledge Dr. Ruth Douglas Miller from the EE department for always keeping her door open to me for any and all questions.

Finally a special thanks to my office mates Joshua Cook and Faryad Darabi Sahneh for their continuous moral and technical support.

CHAPTER 1 - Introduction

Wind turbines provide the energy industry with a clean alternative to burning fossil fuels in order to generate electric power. Wind turbines are a zero-emission machine capturing the renewable energy provided by wind, ultimately originating from the sun. There are all types of different designs, sizes and implementations of wind turbines due to factors in the site-specific application and wind condition. The basic design of a wind turbine can be classified into one of following two categories: vertical axis wind turbines (VAWTs) and horizontal axis wind turbines (HAWTs), although other contemporary turbine designs exist [1]. Figure 1.1 is a diagram of each of these types of turbines.

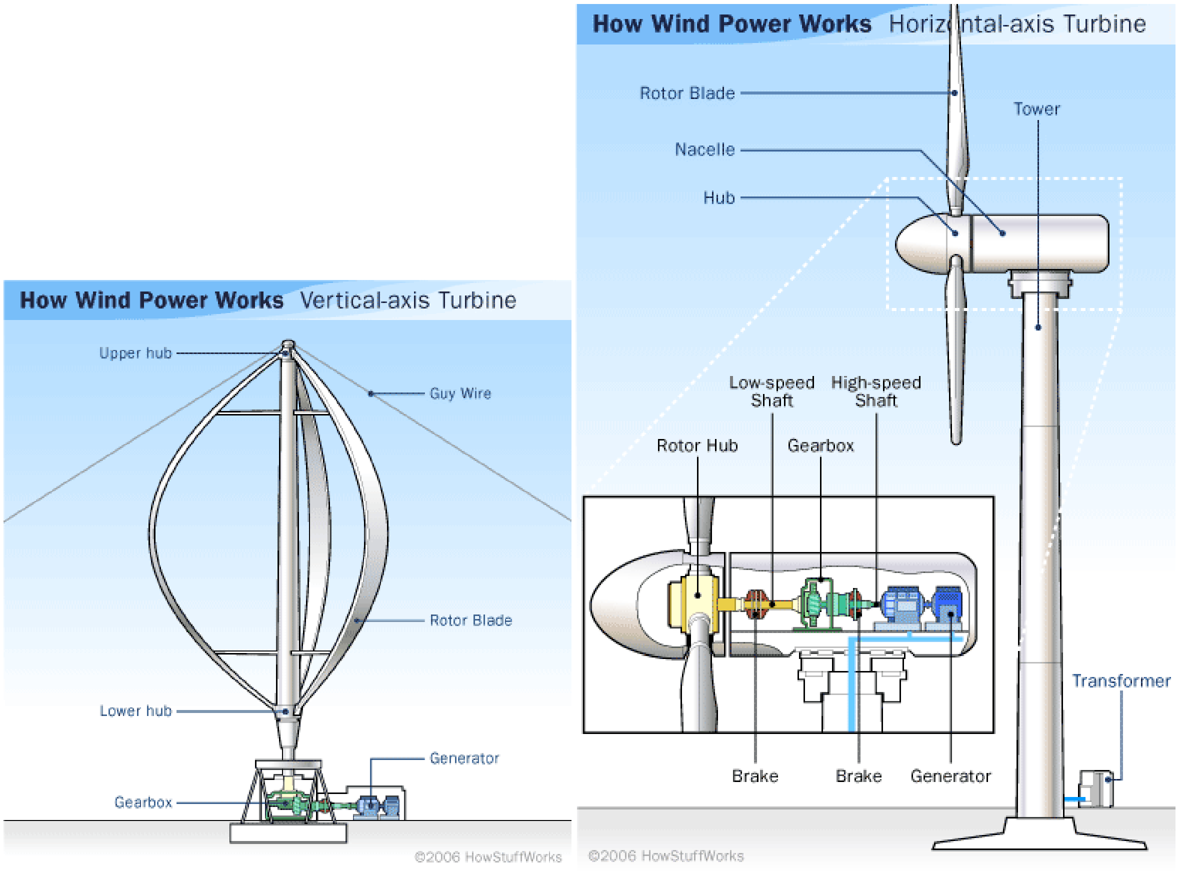

Figure 1.1 Vertical axis (VAWT),left, and horizontal axis (HAWT), right, design configurations for wind turbines. [Diagram Courtesy of [2]]

There are different advantages and disadvantages to each design. For example, horizontal-axis wind turbines are able to avoid energy loss because the blade motion is always perpendicular to the wind movement; however, this design also requires the elevation of heavy components such as gear boxes and the generator. VAWTs are able to capture wind energy in areas with frequent shifts in direction such as urban environments. They also allow the heavy machinery to be located at the ground level, but vertical axis designs must endure more mechanical stresses due to the asymmetry of the forces placed on the rotor blades [3]. This thesis will consider turbines of HAWT design because they are the most commonly implemented configuration for commercial energy production.

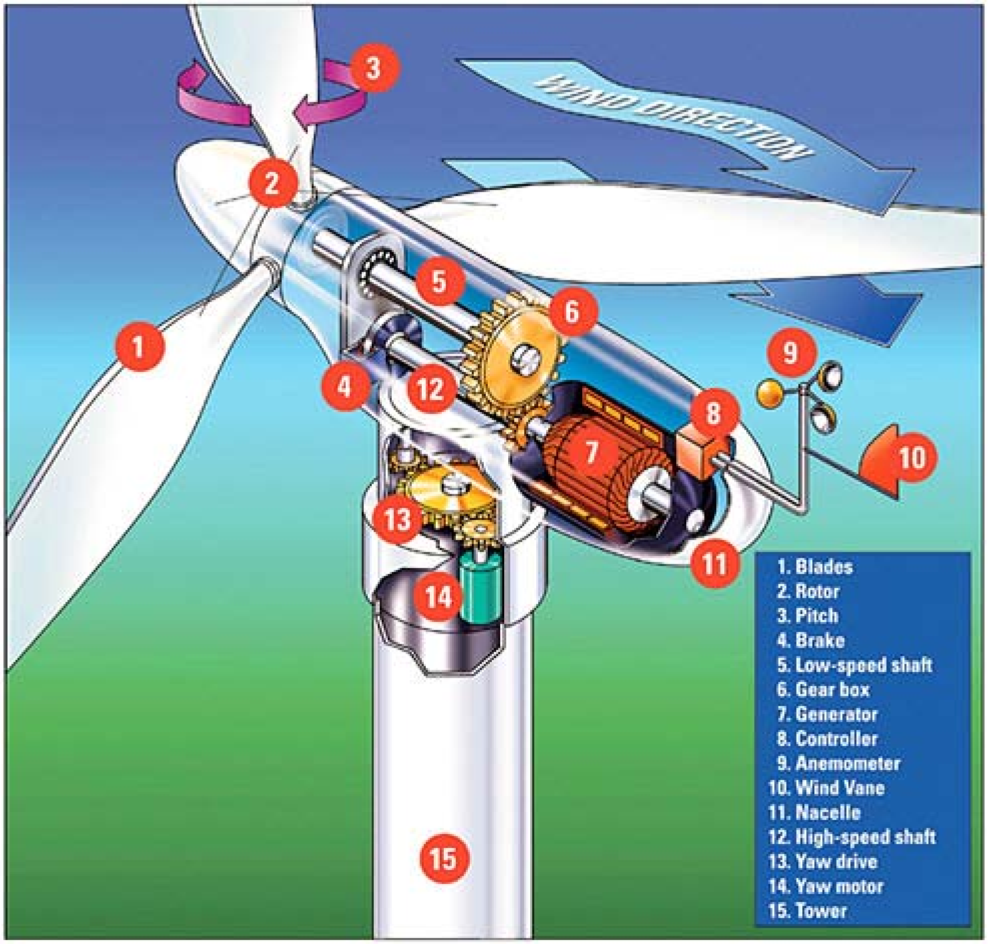

Figure 1.2 Breakdown diagram of a HAWT and all of its working components. [Diagram Courtesy of [4]]

Figure 2.2 shows all of the components of a typical horizontal-axis wind turbine. The wind moves across the rotor blades providing a torque to a primary drive shaft known as the low-speed shaft. This shaft then goes to a gear box which is geared up so that the output shaft is spinning faster. This output shaft is known as the high-speed shaft and is connected to the rotor of the generator. The generator then turns to generate electrical power which is then sent down the tower of the wind turbine via wiring. The yaw motor is used to point the wind turbine in the direction of the prevailing wind. The brake is used to provide additional damping to the low speed shaft. Two important measurement devices are the wind vane and anemometer which sit on the back of the nacelle in this design. These devices are used to measure the wind direction and wind velocity respectively.

The size of the wind turbine is also greatly determined by the application. Many private wind turbines are owned by schools, businesses, and residences that are rated on the order of kilowatts. These small scale wind turbines are attractive because many states offer grants, subsidies, tax benefits, and utility buyback of excess power produced. Smaller turbines also have the benefit of lower cost, simplistic design for low maintenance, and operation in low wind speed. In California, a 10-kW home wind system costs about $16, 000 to install after the state rebate, and produces an average of 900–1500 kWh of electricity per month. It is reported that the homeowners get their return on investment in six to ten years [5].

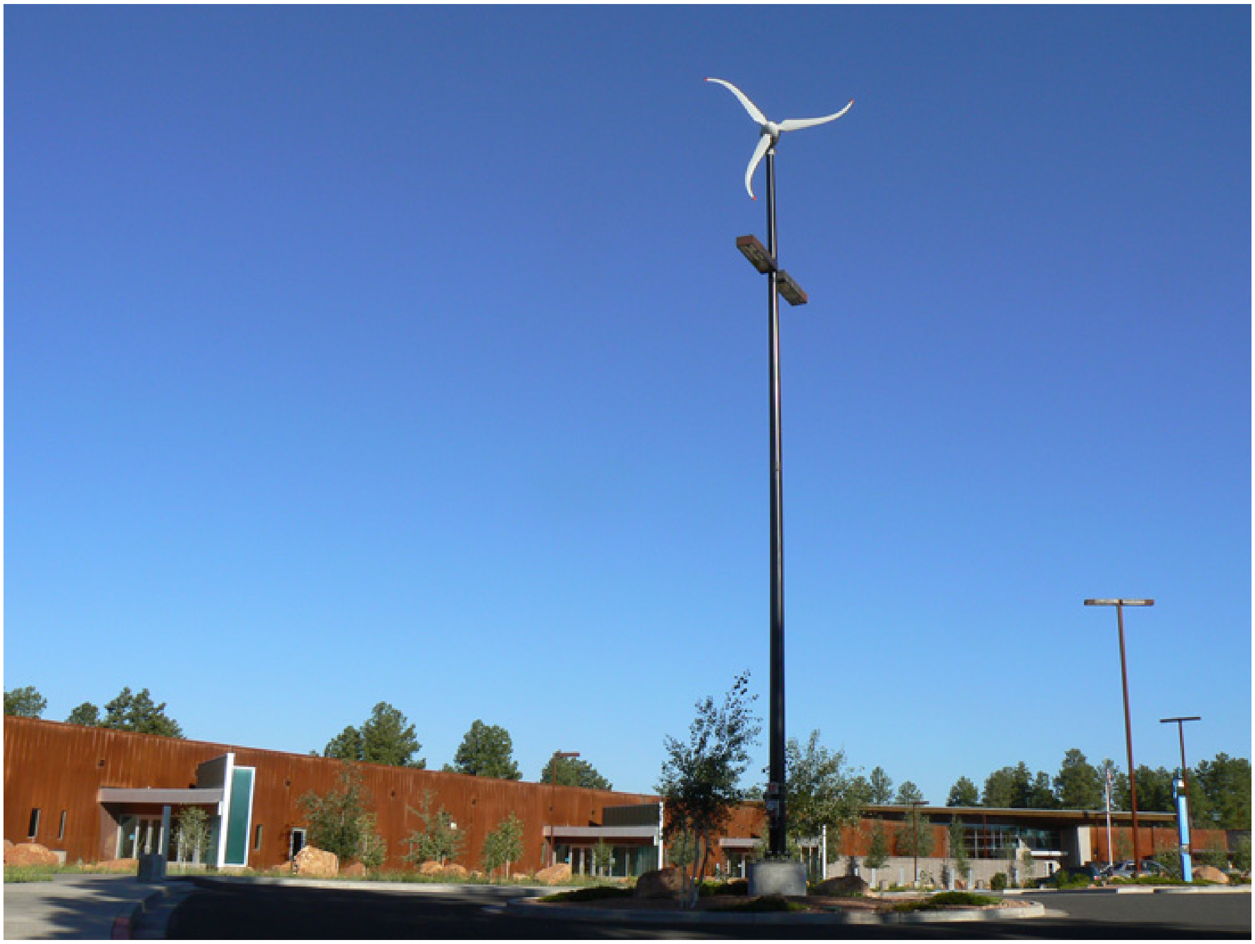

Figure 1.3 Example of the a Skystream wind turbine installed at Coconino Community College. [Photo Courtesy of [6]]

Small wind turbines are also being implemented for use in schools. Figure 1.3 shows a turbine manufactured by Skystream for these applications. The *Wind for Schools* program coordinates the erection of turbines in K-12 schools for the purpose of energy generation as well as educational use in which the students can use real data to learn about wind capture [7].

While small-scale turbines are practical for private use, the commercial production of energy at the utility scale requires much larger wind turbines. These large wind turbines have generating capacity of 850kW-3.0MW and rotor-swept diameters ranging from 52-112 meters according to a review of commercial turbines produced by Vestas [8]. These large turbines employ highly engineered airfoil designs, mechanical components, and electrical components with the purpose of producing the greatest amount of energy possible. Competition is growing among commercial-scale turbine manufacturers with the number tripling since 2005. GE leads the market in number of wind turbines and capacity installed, producing just less than 4,000MW, with its nearest competitor, Vestas at just under 1,500MW installed [9]. In addition to inland turbines, commercial harvest of wind has looked to installing turbines for offshore applications, shown in Figure 1.4. Wind speeds offshore tend to be higher and more consistent. Another benefit to offshore installations near large cities and load centers is the limited development of inland turbines and high real estate value, [10].

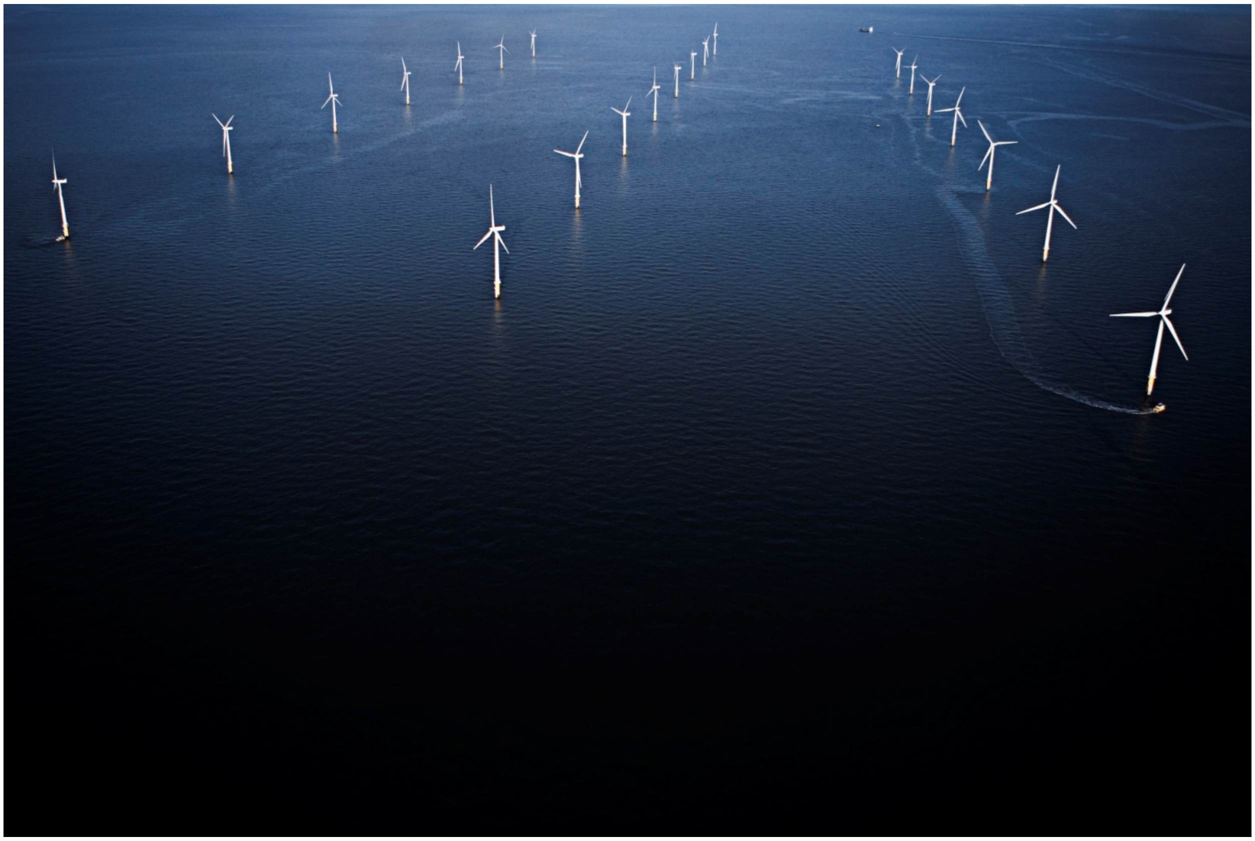

Figure 1.4 Offshore wind turbines typically are a more reliable power source due to the steady and consistent nature of wind over the ocean. Shown is an offshore wind project using GE 2.5MW offshore-design turbines. [Photo Courtesy of [11]]

Wind turbine use has increased drastically throughout the past decade, [12]. Figure 1.5 below shows the world installed wind energy capacity. The increase in wind turbine use has driven researchers to examine how the device captures power and how they can improve its energy efficiency.

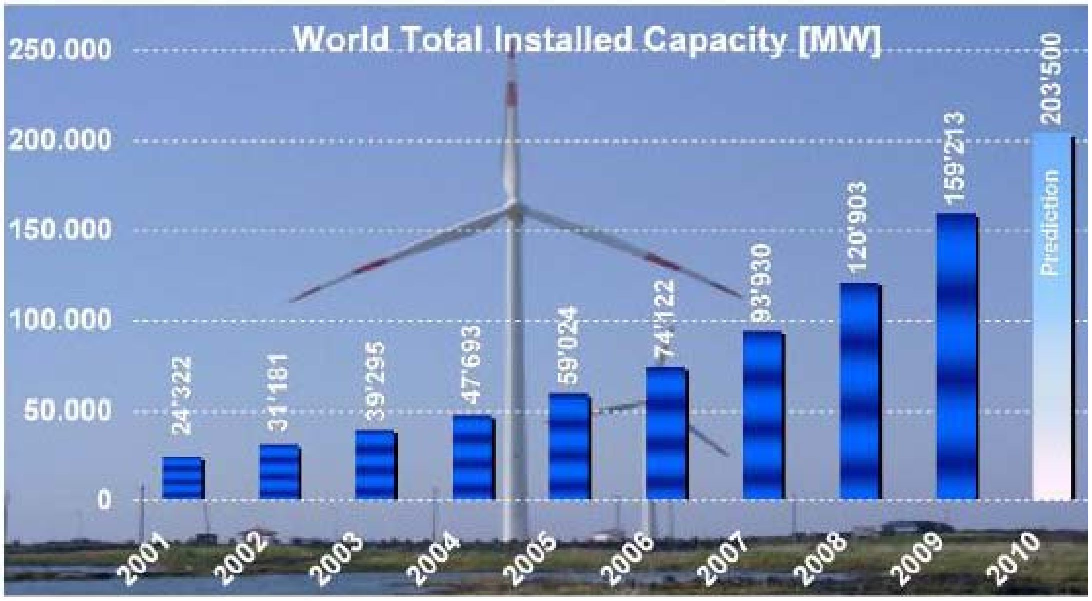

Figure 1.5 World Wind Energy Installed Capacity [Graph Courtesy of [13]]

There are three major stages of wind energy capture: (1) The mechanical system's capture of wind power, (2) the conversion of mechanical energy to electrical energy, and (3) the integration of the converted electrical energy into the power grid [14]. The first and second stages of energy conversion offer the most opportunity for improvement; therefore, this thesis will cover the first stage, improving the conversion of wind power to mechanical energy.

There are three wind conditions in which a turbine operates. In Region I, the wind velocity is below the cut-in wind speed of a turbine that is starting up. When the wind speed is very low, it is not feasible to operate the turbine for power capture. When operating in Region I, the turbine captures no wind power and does not produce any electrical energy. In Region III, the wind velocity is so great that it requires the rotor angular velocity to extend beyond the safe operating range. In this region, the power capture is (should be) limited below its optimal efficiency to ensure the mechanical and electrical loads are not exceeded [15]. The wind turbine produces power in Region III until wind velocity becomes so great that it reaches the cut-out wind speed and turbine operation ceases completely to avoid damaging the system.

Region II is the mid-speed, wind operating range. Here, wind velocities are below the rated wind speed and above the cut-in speed. This is the region where the turbine spends most of its time operating. In Region II, the turbine power capture is limited only by its efficiency. The scope of this thesis is wind turbines operation in Region II. In Figure 1.6 is a graph of the power capture in the three wind regions.

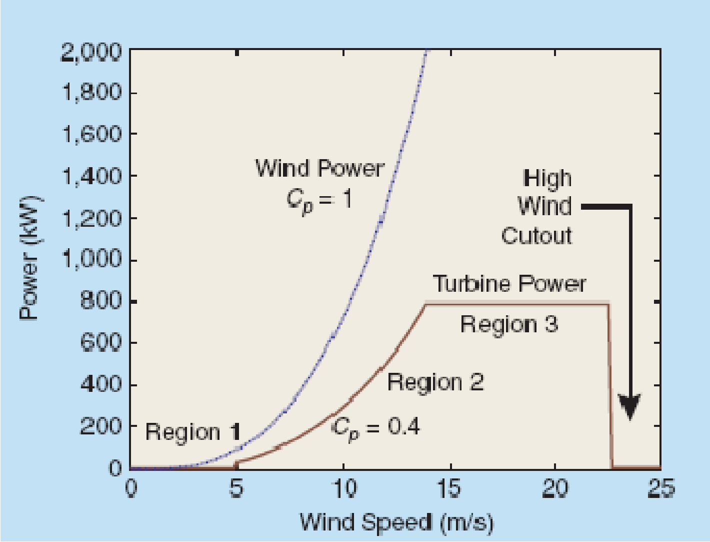

Figure 1.6 The three regions of operation of a wind turbine due to wind velocity. [Diagram Courtesy of [15]]

In addition to the variety of airfoil designs and generator sizes, there are many variations on drive train mechanism configurations: DC generators, AC synchronous generators, direct drive shafts, turbines with gear boxes, fixed pitch, variable pitch, and asynchronous generators are some of the designs that have been employed. One of the most prominent variable speed generators is a doubly-fed induction generator (DFIG) [16].The DFIG is more popular than other variable speed generator designs because the electronics handle only a fraction of the power compared to a full-scale power converter required for other designs[17].

Many of the modern wind turbines used today for electricity production are variable-speed and variable-pitch turbines. Unlike a fixed-speed turbine, the rotor of a variable-speed turbine can vary its speed to follow wind patterns. This important quality allows the turbine to vary its angular velocity and maintain a tip-speed ratio which yields maximum power production in gusty and changing wind conditions. The variability of the blade pitch is typically used to limit power capture beyond the rated capacity of the turbine when there are high wind speeds. Pitch control can also be used to control maximum power output because of its relationship to the capture of the wind energy. Because of the advantages of this type of wind turbine, this thesis will consider the application of the proposed control method to a variable speed, variable pitch wind turbine. Operation of a variable speed, variable pitch wind turbine requires a complex control system to alter the pitch and generator loading. One requirement of the control system is to automatically regulate the wind turbine to some specified operation. One other, and more difficult, use of automatic control is to specify the operation of the turbine at its peak operating efficiency.

Wind turbine automatic control which incorporates maximum energy capture has proved challenging. This is because the power provided to the turbine by the wind/rotor interaction has a difficult-to-measure nonlinear aerodynamic property. This power capture characteristic is a function of the wind velocity, rotor angular velocity, and blade pitch. This property is known as the coefficient of performance and is unique for each wind turbine in shape and optimal peak location. To extend the challenging nature of the problem, the optimal operating point is not always common between similar machines operating under the same conditions. The optimization of wind capture efficiency is not only a problem of nonlinear system trajectory tracking control, but also estimation and extremum seeking. One other aspect that a wind turbine control should address are the changes of time varying system parameters. In a practical wind capture system, changes such as component aging, oxidation, icing, and friction will cause a variation in the optimal peak location and shape of the power coefficient relationship.

Review of Literature of Wind Turbine Control for Maximum Power

Maximizing power capture for variable speed wind turbines in Region II by means of control theory has been popular topic for researchers within the past one and a half decades. Many engineers have approached this problem by applying different control methodologies. Using blade pitch control, generator torque control, or a combination of both, automatic control systems have been built into wind power capture systems to stabilize their operation and maximize power capture. A good overview of wind turbine control methods can be found in [18].

Linear control methods have been used to determine a linearized model of the wind turbine dynamic system [19]. Such approaches use PID (proportional-integral-derivative) or similar controllers to regulate the system states. State feedback gains are then tuned to each wind turbine system. These linear control methods assume *a priori* knowledge of the optimal operating point. This is not practical however, because the optimal operating point is not measurable and is different for every turbine. This control method would require experience in a particular turbine's behavior and manual re-tuning in the event of any parameter variation such as icing, aging of system components and so forth.

Different implementations of sliding mode control have also been used as a nonlinear control approach. In [20-21], sliding mode controllers were proposed for a system in order to provide tracking in the presence of the unknown nonlinear input torque. The results of this control method provide better tracking control than linearized PID controllers. However, the sliding mode control method has been known to induce chattering in the system due to the switching nature of the controller. This is not good because the chattering along the desired trajectory is indicative of system vibration. These strategies also rely on measurement of the captured power which is a parameter that is generally considered not reliably measureable [28].

Researchers have also developed adaptive techniques to aid in identifying unknown quantities. Perhaps one of the greatest challenges that wind turbine control presents is the lack of available information. In fact, the coefficient of performance, which is intended to be maximized, is a quantity that is not readily measured or calculated. Johnson et al. proposed a nonlinear controller, [15] and [22], where an adaptive state feedback gain is tuned based on an estimation of the power captured over a period of time. This approach performed well in the convergence toward the optimal operating region and the nonlinear control law was very simple. However, the average power estimation algorithm had a very long adaptation period and convergence time to a widely bounded region of optimal operation. This theory did not take into account blade pitch controllability, which simplifies this method to a single-dimensional optimization problem.

Other estimation approaches have also been studied such as Ma's wind turbine controller given in [23]. A least-squares Kalman filter approach is used to estimate the unknown aerodynamic torque nonlinearities while a proportional-integral controller regulates the turbine to a desired set point. This method provides a good method of smooth regulation of the system states using a Kalman filter. However, there are drawbacks to this method, including lack of a technique for extremum seeking, an assumption that the tip-speed ratio corresponding to the optimal performance coefficient is known, and no consideration of blade pitch control.

Other nonlinear robust strategies have been developed to estimate the nonlinearities. Iyasere et al. [24-25] use a robust controller to provide an estimation of the unknown aerodynamic torque while employing Powell's optimization method to provide a reference trajectory to follow. The result of this method was good tracking of the desired state as well as convergence to the optimal operating point. This method also used both rotor speed and blade pitch control to maximize energy capture. Although this method is a good application of robust control, it lacks practicality because of the assumption of operation under a constant wind velocity.

Extremum seeking control methods have also been used in the application of wind turbine power maximization. Creaby et al., in [26], use an extremum seeking method which is developed from the theory of Ariyur and Kristic [27]. The result of this method is good convergence to the optimal operating point under the conditions of a time varying wind velocity. This method also uses both pitch and torque control. One drawback of this method is the fact that it relies on oscillatory perturbation of the optimizing variables. For the specific application to large wind power capture systems, this may introduce unwanted mechanical vibration and the associated fatigue, thus exacerbating the already high wind turbine maintenance needs. Also, it may not be practical to perturb large inertia systems with high frequency disturbances due to lengthy system time constants.

The control approach presented in the current work uses a Lyapunov-based strategy for maximizing the power capture given time varying wind conditions. This approach will control the blade pitch and shaft speed of a large-scale wind turbine (greater than 1.0 MW) so that the turbine is operating very near the peak power capture. The controller has two responsibilities. The first is regulating of the turbine shaft angular velocity to a given set point. The second is providing the regulator with a set-point trajectory which will drive the blade pitch and the tip-speed ratio in the direction of improving the rotor power coefficient. System parameter changes and time-varying wind velocity are present in practical wind power capture systems. The Lyapunov-based method will ensure the convergence of the power coefficient to a region near its optimal value in the presence of a time varying real wind condition and measurement error.

This presentation will first introduce the elements of a wind turbine energy capture system. The focus will be the description of the power capture coefficient, tip-speed ratio, and the wind turbine equation of motion. Next will be the theoretical development of the robust controller used for regulation and estimation. The next section is the Lyapunov-based control theory development. This section will also include techniques used to reduce the effects of error on system performance. A modeled wind turbine will then be simulated and the results will be reported and discussed in detail to demonstrate the performance of this control method. Finally, a conclusion of the results will be given, with a discussion of work to be done in the future of this research.

CHAPTER 2 - Wind Turbine Dynamics

Tip-Speed Ratio and Blade Pitch

The two controllable parameters of the wind capture system are the tip-speed ratio and the blade pitch. The tip-speed ratio can be controlled by altering the angular velocity of the drive shaft. The tip-speed ratio of a wind turbine can be defined by the following relationship between the angular velocity of the rotor and the wind velocity, which is

$$\lambda = \frac{\omega R}{v}, \tag{1}$$

where λ is the tip-speed ratio, ω is the measurable time-varying rotor angular velocity, R is the radius of the circle swept by the blades, and v is the measureable time-varying wind velocity $v(t)$. In this document it shall also be the convention to write the time varying $\omega(t)$ as ω, as is done with v. Because all of these parameters are either known or measurable, the tip-speed ratio is a known quantity. The goal is to have λ controlled so that it approaches its optimal value, denoted by λ^*.

The blade pitch is denoted by β. The blade pitch is a controllable parameter of the wind turbine. The pitching of the blades is typically done by different servo motors operating for each of the turbine's blades. In this thesis it can be assumed that value of the blade pitch will be common between all of the blades of the turbine. It is also important to note that there are some dynamics associated with the actuation of the pitch command. For this application, it is assumed that the actuation time constants are small compared to the time constants of the control loops; therefore, in this thesis these blade pitch actuation dynamics can be neglected.

The two loops of the proposed control method will contain state variables. Each control loop will have two state variables. The inner control loop, responsible for regulation of the drive shaft angular velocity, will have the shaft angular velocity, ω, as one of the state variables. The second state variable of the inner closed loop system is an error term, $r(t)$, which will be developed later in Chapter 3. The outer control loop also contains two state variables. The outer loop control objective is to regulate λ and β to their optimal values, so naturally, λ and β are the state variables for the outer loop.

Power Capture Coefficient

The power capture coefficient, also known as the coefficient of performance, is denoted by C_p. C_p is an unknown, nonlinear, and unmeasureable parameter. The coefficient of performance is a function of λ and β. Because the tip-speed ratio and blade pitch are both controllable parameters, C_p is completely controllable. C_p is defined as the ratio of the power captured by the wind turbine to the power available from the wind. This relationship can be expressed as

$$C_p(\lambda,\beta) = \frac{P_{cap}}{P_{avail}}. \tag{2}$$

P_{cap} is the mechanical power captured by the turbine and P_{avail} is the power available from the wind. As seen in Eq. (2), P_{cap} will be maximized when C_p is maximized for a particular wind speed. This can be shown by expressing the available power, P_{avail} in terms of the wind velocity,

$$P_{avail} = \frac{1}{2}\rho A v^3, \tag{3}$$

where A is the circular area swept by the known blade radius R. The term ρ is the air-mass density. These parameters are considered to be time-invariant and known. The exception to these is the time-variability of the air density. It is considered to be very slowly time varying compared to the dynamic system, thus will not interfere with convergence and/or the stability of the controller; therefore, this we will consider it as a constant. Because all of these parameters are known and the wind velocity is measurable, the term P_{avail} is a known quantity.

The term P_{cap} is considered to be unknown and will be estimated. It does not however seem that P_{cap} should be treated as unknown because of the apparent availability to measure P_{cap} by the generator-produced electrical power. First, the power captured by the generator experiences some factor of loss due to the conversion of mechanical energy to electrical energy. Secondly, the power measurement of the generator is unreliable due to the high variability [28]. Here, P_{cap} is considered to be unknown because of these factors. It will also add an estimable unknown quantity to the controls problem. The expression for P_{cap} can be defined as

$$P_{cap} = \frac{1}{2}C_p(\lambda,\beta)\rho A v^3, \tag{4}$$

It can be seen from (4) that the reason for P_{cap} being unknown is that it contains the unknown coefficient-of-performance term, $C_p(\lambda,\beta)$.

According to aerodynamic blade-moment theory, the maximum possible value of C_p is 0.59, known as the Betz limit [29]. This limit is the maximum theoretical power capture ratio.

Most turbines are capable of achieving power capture coefficients in the typical range of 0.4-0.5 at the peak of their C_p curves due to losses in the physical system. In practice, most wind turbines actually capture power in the range of 0.3-0.4 due to the controllers. Figure 2.1 shows a graphical representation of a typical three dimensional C_p curve as a function of λ and β. (Plot data acquired from the Simulink Wind Turbine Blockset, [33])

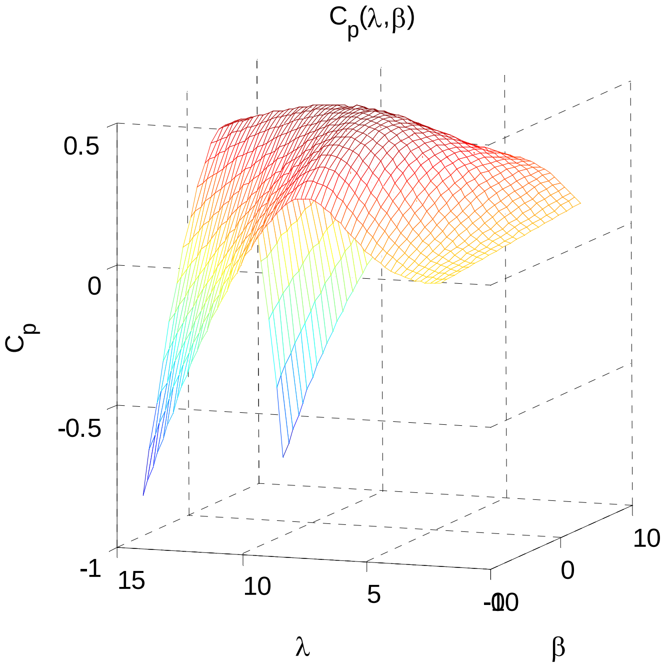

Figure 2.1 Coefficient of performance, $C_p(\lambda,\beta)$, in the realistic operating region of λ and β. C_p has a maximum value of $C_p^* = 0.4725$ which corresponds to this particular turbine's ability to capture wind → mechanical power at 47.25% efficiency.

In Region II, the power capture is limited by C_p because the wind velocity is above the cut-in and below the Region III wind velocity range where the power capture is intentionally limited to avoid excess angular velocity. The turbine is converting wind energy to mechanical energy at its optimal level when the power coefficient C_p is equal to C_p^*. C_p^* is the maximum or extremum value of C_p atop the curve. C_p^* is always greater than zero and is considered to be constant or slowly time varying. The values of λ and β which correspond to the value of C_p^* are given as λ^* and β^*. It is therefore desirable, in Region II, to control parameters λ and β so that they operate at these values of λ^* and β^* so that the power coefficient is maximized, operating at C_p^*, thus attaining the maximum possible wind energy capture.

Differential Equation – Inner Loop Dynamics

The wind turbine considered in this paper is a direct-drive, single-mass system. Using this model type helps to simplify the turbine dynamics by lumping inertias together around the rotational axis of the drive train. This allows the rotational inertias of the rotor cone, blades, drive shaft, electric generator, and anything else in motion about the axis of rotation to be represented by the single term *J*.

The same concept is applied to the viscous damping which occurs along the drive train rotational axis. All viscous damping is combined into the single parameter C_D. The viscous damping is multiplied by the state to indicate that it is proportional and always opposes the direction of angular motion giving C_D its negative sign (see Eq. (5) below). The open-loop wind turbine dynamic system is described by the differential equation

$$\dot{\omega} = \frac{1}{J}(\tau_{aero} - C_D\omega - \tau_c). \tag{5}$$

ω is the rotor shaft angular velocity, τ_{aero} is the aerodynamic torque provided to the drive shaft from the wind interacting with the rotor blades, and τ_c is the reaction torque the drive shaft gets from the generator. $\dot{\omega}$ is the first time derivative of the shaft angular velocity which corresponds to the shaft angular acceleration.

Of the parameters presented here, it is considered that *J* and C_D are known, constant, and time-invariant. τ_{aero} and $\dot{\omega}$ are unknown and unmeasurable time-varying parameters. τ_c is the control input to the dynamic system. This controllable torque is provided via generator loading which produces a torque opposing the motion of the aerodynamic rotor torque. This loading of the generator can be altered to provide the desired τ_c.

Although τ_{aero} is unknown, it can be described from the following relationship,

$$\tau_{aero} = \frac{P_{cap}}{\omega}. \tag{6}$$

By substituting (3) into (6), τ_{aero} can be represented as

$$\begin{aligned} \tau_{aero} &= \frac{1}{2}\frac{C_p(\lambda,\beta)}{\omega}\rho A v^3 && (7) \\ &= \frac{1}{2}\rho A R \frac{C_p(\lambda,\beta)}{\lambda} v^2 && (8) \\ &= \frac{1}{2}\rho A R C_q(\lambda,\beta) v^2 && (9) \end{aligned}$$

where

$$C_q(\lambda,\beta) = \frac{C_p(\lambda,\beta)}{\lambda}. \tag{10}$$

These equations, (1-10), make up the equations used to model the behavior of the wind turbine system. The goal of the wind turbine automatic control system is two-fold. 1.) Ensure stability of the system state, ω. 2.) Control λ and β in such a way that C_p is maximized. To meet our desired control objectives, a control law for τ_c must be developed to regulate the angular velocity of the drive shaft to maintain a some desired λ. A control law must be written for λ and β so that $C_p \rightarrow C_p^*$.

Closed Loop Dynamic System – Control Strategy

The control strategy will be broken up into two loops. First, the inner loop will control the closed loop wind turbine dynamic system given in (5) by varying the term τ_c. This controller will provide the asymptotic tracking of the state ω to some desired trajectory, ω_d. The inner loop controller will also be responsible for estimating the aerodynamic torque, τ_{aero}. Second, the outer loop will control the parameters λ and β to ensure their convergence to their values of λ^* and β^*. This will provide a trajectory path, ω_d, for the inner loop control to track. Figure 2.2 shows a block diagram of the proposed control system.

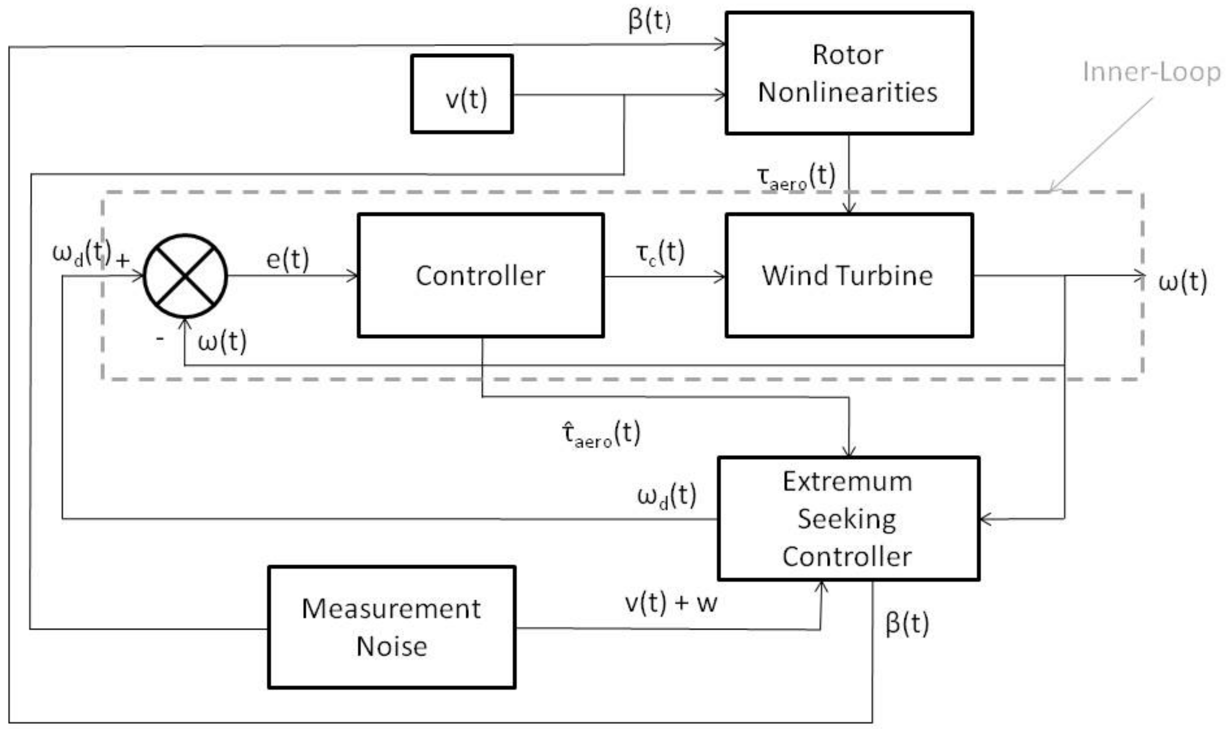

Figure 2.2 Two-loop control block diagram of proposed control method. The inner loop (robust controller) regulates the wind turbine shaft speed. The outer loop (Lyapunov-based controller) controls the ω_d and blade pitch, β, values.

CHAPTER 3 - The Nonlinear Robust Estimator Controller

A nonlinear robust controller is developed to regulate the state of the system, ω about a desired set-point, ω_d. This is otherwise known as a tracking controller. As a result of the asymptotic tracking control, an estimation of the unknown parameter τ_{aero} is also obtained. This estimation of the aerodynamic torque is denoted as $\hat{f}$. A reaction torque from the generator, τ_c, is used to control the angular shaft velocity of the system. Recall from (5) that the dynamic equation of the wind turbine is given by,

$$\dot{\omega} = \frac{1}{J}(\tau_{aero} - C_D\omega - \tau_c).$$

where τ_{aero} is an unknown and time-varying nonlinear term in the dynamics, which contains the term $C_p(\lambda,\beta)$ as shown in (8) and (9). Equation (5) can be rewritten as

$$J\dot{\omega} + C_D\omega - \tau_{aero} = -\tau_c \tag{11}$$

The objective of this identifier-based control design is twofold:

(1) Achieve asymptotic tracking in the sense of $\omega(t) \rightarrow \omega_d$ where ω_d represents the desired angular velocity.

(2) Estimate the unknown nonlinear function τ_{aero}.

Development of the Robust Estimator Controller

A tracking error of this desired trajectory is defined as

$$e = \omega_d - \omega. \tag{12}$$

To facilitate the subsequent control design and analysis, a filtered tracking error, denoted as $r(t)$, is defined as

$$r = \dot{e} + \alpha e \tag{13}$$

where α denotes a positive constant. The filtered tracking error $r(t)$ is not measurable because it depends on $\dot{\omega}(t)$ which is also not measurable. Multiplying (13) by J gives the following

$$Jr = J\dot{\omega}_d + J\alpha e + C_D\omega - \tau_{aero} + \tau_c \tag{14}$$

where (5), (12) and (13) were utilized. Based on the expression in (14), the control torque is designed as

$$\tau_c = -J\dot{\omega}_d - J\alpha e - C_D\omega + \hat{f}(t) \tag{15}$$

where $\hat{f}(t)$ denotes a subsequently designed control term. By substituting (15) into (14), the result becomes

$$Jr = -\tau_{aero} + \hat{f}(t). \tag{16}$$

From (16) it is evident that as $r(t)\rightarrow 0$, $\hat{f}(t)$ will identify the unknown input torque τ_{aero}; therefore, it is desirable to design a controller such that $r(t)\rightarrow 0$. To facilitate the design of $\hat{f}(t)$, we differentiate (16) which gives

$$J\dot{r} = -\dot{\tau}_{aero} + \frac{d\hat{f}(t)}{dt}. \tag{17}$$

Based on (17) and the analysis found in appendix A, the control law $\hat{f}(t)$ is designed as

$$\begin{aligned}\hat{f}(t) = (k_s + 1)e(t) - (k_s + 1)e(0)\\ + \int_0^t [(k_s + 1)\alpha e(\tau) + \beta_c \operatorname{sgn}(e(\tau))] d\tau\end{aligned} \tag{18}$$

where k_s, α and β_c are positive constant control gains. The sgn(·) denotes the standard signum function. [Note that α was defined previously in (13).] The time derivative of (18) is given by

$$\frac{d\hat{f}(t)}{dt} = (k_s + 1)r + \beta_c \operatorname{sgn}(e(t)). \tag{19}$$

After substituting (19) into (17), the following closed-loop error system can be obtained as

$$J\dot{r} = -(k_s + 1)r - e - \beta_c \operatorname{sgn}(e(t)) + N \tag{20}$$

where the auxiliary function N denotes the unmeasurable auxiliary term of

$$N(\omega, \dot{\omega}, t) = -\dot{\tau}_{aero} + e. \tag{21}$$

Before analyzing the stability of the closed-loop system, we perform the following manipulation on (21). Let us introduce a new, unmeasurable parameter N_d that is defined as

$$N_d(\omega_d, \dot{\omega}_d, t) = -\dot{\tau}_{aero}. \tag{22}$$

The reason to introduce N and N_d is to facilitate the stability analysis. The main result can be stated in the following theorem.

Theorem 1: The controller given in (18) and (15) achieves semi-global asymptotic position tracking in the sense that

$$e(t) \rightarrow 0 \text{ as } t \rightarrow \infty$$

provided that β_c is selected according to the sufficient condition

$$\boldsymbol{\beta_c} > |-\dot{\boldsymbol{\tau}}_{\boldsymbol{aero}}| + \left|\frac{-\ddot{\boldsymbol{\tau}}_{\boldsymbol{aero}}}{\boldsymbol{\alpha}}\right|.$$

(Proof for β_c selection criteria found in [31, 32])

In addition, all system signals are bounded, and τ_{aero} can be identified in the sense that

$$\left|\hat{f}(t) - \tau_{aero}\right| \rightarrow 0 \text{ as } t \rightarrow \infty$$

The stability analysis of this controller is provided in Appendix A.

CHAPTER 4 - The Lyapunov-Based Extremum Seeking Controller

Using the Estimate of $\hat{f}$ to Estimate C_p

Using the stability analysis of the robust estimator controller, it has been shown that $\hat{f}(t) \rightarrow \tau_{aero}$ as $t \rightarrow \infty$. Let $\hat{f}$ be the estimate of τ_{aero}. Before the estimate, $\hat{f}$, should be used to approximate τ_{aero}, it must be certain that $\hat{f}$ is accurate within some tolerance region. From (16), as $r(t) \rightarrow 0$ then so must $e(t) \rightarrow 0$. Therefore, when $e(t) \leq \epsilon$, the estimate is said to be within the specified error tolerance, where ϵ is some positive constant error tolerance chosen by the designer. This ensures that the estimate, $\hat{f}$, is very close to the actual value of τ_{aero}. Now that we have a very close approximation to τ_{aero}, $\hat{f}$ may be used to estimate our power performance coefficient C_p. The estimate of C_p is denoted as $\hat{C}_p$ and defined utilizing the relationships given in equations (1) and (6). Our equation for expressing the estimated C_p is given by

$$\hat{C}_p = \frac{\hat{P}_{cap}}{P_{avail}}, \tag{23}$$

where

$$\hat{P}_{cap} = \hat{f}\omega, \tag{24}$$

and P_{avail} is given in (2). The reason that $\hat{f}\omega$ may be used to describe $\hat{P}_{cap}$ can be seen from the relationship in (6). The captured power can be written as a product of the angular shaft velocity, ω, and the rotor torque τ_{aero}.

The robust nonlinear estimator controller provides the wind capture system with two things: The ability to track a desired trajectory and an estimate of the unknown nonlinear time-varying rotor torque. Using this torque estimate we can provide the outer loop controller with an estimate of C_p which is finally stated as

$$\hat{C}_p = \frac{\hat{f}\omega}{\frac{1}{2}\rho A v^3}. \tag{25}$$

Equation (25) will then be used by the outer loop controller to manipulate states λ and β so that it is maximized.

Lyapunov Candidate Function – Theoretical Development

Previously, we have developed a controller which ensures tracking of a desired trajectory for inner-loop state variable ω. It is now time to develop the outer-loop control law which will provide a trajectory to follow. It is desirable that a trajectory be generated which ensures the turbine's operation at the point of optimal power capture, C_p^*. Thus, we provide a control law which drives our two outer-loop system states, λ and β, to their optimal values, λ^* and β^*, respectively.

To ensure convergence to the optimal point, a Lyapunov function is developed as

$$V = \frac{1}{2}\tilde{C}_p^2 > 0, \tag{26}$$

where

$$\tilde{C}_p = C_p^* - C_p \geq 0, \tag{27}$$

the error in our power coefficient from its optimal value. The time derivative of our Lyapunov function is given by

$$\dot{V} = -\tilde{C}_p \dot{C}_p. \tag{28}$$

Here $-\dot{C}_p$ by way of (27), may be substituted for $\dot{\tilde{C}}_p$ because the optimal point is considered to be constant or slowly time varying. This is a permissible assumption because the time varying nature of C_p^* is due to aging or climate effects which will be greatly and consistently slower than any time constants in our control law or the dynamic system. Because C_p^* is then not a function of time, differentiating $\tilde{C}_p$ leaves then only $-\dot{C}_p$. The definition of $\dot{C}_p$ can be expanded to

$$\dot{C}_p = \frac{\partial C_p}{\partial \lambda}\dot{\lambda} + \frac{\partial C_p}{\partial \beta}\dot{\beta}. \tag{29}$$

The final time derivative of the Lyapunov function becomes

$$\dot{V} = -\tilde{C}_p\left(\frac{\partial C_p}{\partial \lambda}\dot{\lambda} + \frac{\partial C_p}{\partial \beta}\dot{\beta}\right) \quad \leq 0. \tag{30}$$

To begin our analysis, we must be certain that $\dot{V} \leq 0$ to prove that $C_p \to C_p^*$. We can be sure that $\tilde{C}_p$ is greater than zero for all points other than zero because of (27). Recall that C_p^* is defined as the maximum value that C_p can attain. Therefore, $\tilde{C}_p$ can only be greater than zero for any value of C_p other than $C_p = C_p^*$. In this case $\tilde{C}_p = 0$, which means that C_p is at its optimal value and the control goal has been achieved.

The second part of the analysis is to show that

$$\left(\frac{\partial C_p}{\partial \lambda}\dot{\lambda} + \frac{\partial C_p}{\partial \beta}\dot{\beta}\right) \geq 0. \qquad .$$

Recall that λ and β are controllable parameters of the wind turbine. This means that an update law for $\dot{\lambda}$ and $\dot{\beta}$ can be selected. The update laws are designed as

$$\dot{\lambda}_d = \gamma_\lambda \mathrm{sgn}(\frac{\partial C_p}{\partial \lambda}) \tag{31}$$

and

$$\dot{\beta} = \gamma_\beta \mathrm{sgn}(\frac{\partial C_p}{\partial \beta}), \tag{32}$$

where sgn(·) denotes the standard signum function. The substitution of $\dot{\lambda}_d$ for $\dot{\lambda}$ is used in (31) due to the fact that there are dynamics between the desired tip-speed ratio, λ_d, and the actual λ. These are the dynamics of the system given in (5). λ will be equal to λ_d when $\omega = \omega_d$. This is not done for $\dot{\beta}$, however, because we have chosen to neglect the dynamics in blade pitch.

Because of the requirement on $\dot{\lambda}$ and $\dot{\beta}$ to possess the same sign as $\frac{\partial C_p}{\partial \lambda}$ and $\frac{\partial C_p}{\partial \beta}$, respectively, it can be assured that the term $\left(\frac{\partial C_p}{\partial \lambda}\dot{\lambda} + \frac{\partial C_p}{\partial \beta}\dot{\beta}\right)$ is greater than zero, except for the case where $\frac{\partial C_p}{\partial \lambda}$ and $\frac{\partial C_p}{\partial \beta}$ are equal to zero. According to the nature of a C_p curve, it is a convex function with respect to both independent variables and has a single maximum. The only place where both partial derivatives are equal to zero is when $C_p = C_p^*$. In this case the design goal has been met.

According to this Lyapunov analysis, the state evolution from (31) and (32) ensure that the system will attain its optimal performance coefficient value of C_p^* by driving the term $\tilde{C}_p$ to zero. Because of the unavailability of the C_p function, and thus its partial derivatives, it is impossible to derive an optimization update law which ensures that $\lambda \rightarrow \lambda^*$, $\beta \rightarrow \beta^*$, and $C_p \rightarrow C_p^*$ precisely, however, through estimation, we can achieve bounded convergence.

Discrete Estimation of Partial Derivatives

In the first section of this chapter a method was developed to estimate the unknown power coefficient, given in (25). Recall that this estimate was denoted as $\hat{C}_p$. Recall also that the reliability of this quantity is dependent on some amount of time for $\hat{f} \rightarrow \tau_{aero}$. We are constrained to develop a discrete update law because of this delay in time. Being that C_p is not a measurable parameter, we must also use the estimate of (25) to obtain the partial derivatives, $\frac{\partial C_p}{\partial \lambda}$ and $\frac{\partial C_p}{\partial \beta}$.

To begin our discrete estimation approach we define our discrete time partial derivatives of $\hat{C}_p$ with respect to λ and β. These are given by

$$\frac{\partial \hat{C}_p}{\partial \lambda} \cong \frac{\hat{C}_{p_k} - \hat{C}_{p_{k-1}}}{\lambda_k - \lambda_{k-1}} \tag{33}$$

and

$$\frac{\partial \hat{C}_p}{\partial \beta} \cong \frac{\hat{C}_{p_k} - \hat{C}_{p_{k-1}}}{\beta_k - \beta_{k-1}}. \tag{34}$$

Using these estimates of $\frac{\partial C_p}{\partial \lambda}$ and $\frac{\partial C_p}{\partial \beta}$, our discrete update laws, $\dot{\lambda}_d$ and $\dot{\beta}$, are designed. Based on (31) and (32) the discrete update law equations are

$$\lambda_{d_{k+1}} = \lambda_{d_k} + \gamma_\lambda \mathrm{sgn}(\frac{\partial \hat{C}_p}{\partial \lambda}) \tag{35}$$

and

$$\beta_{k+1} = \beta_k + \gamma_\beta \mathrm{sgn}(\frac{\partial \hat{C}_p}{\partial \beta}). \tag{36}$$

In (35) and (36), $\lambda_{d_{k+1}}$ and β_{k+1} generate the desired trajectories for the states λ and β of the wind turbine outer loop control system. Using (35) and (4), a relationship is developed which generates the desired shaft angular velocity, ω_d, given by

$$\omega_d = \frac{\lambda_{d_{k+1}} v}{R}. \tag{37}$$

The parameter calculated in (37) is the desired angular velocity trajectory for the wind turbine drive shaft. The inner loop controller will use this value of ω_d to regulate the system state, ω, to track it asymptotically. Notice that the term ω_d is a continuous term while $\lambda_{d_{k+1}}$ is a discrete term. Recall that v represents the time-varying wind velocity. Although $\lambda_{d_{k+1}}$ will be constant over the time interval between steps k and k+1, the desired state trajectory will be changing continuously according to the time-varying wind velocity. A critical benefit of this is the ability to maintain a particular value of C_p in a time-varying wind condition, namely, following wind patterns when $C_p = C_p^*$.

Alternating Partial Derivative Computation Method

At this point in our design, a controller has been developed to asymptotically track a desired state trajectory for ω. A method has also been developed estimate the coefficient of performance and select state trajectories of λ and β which maximize the unknown performance coefficient. It is important to note that we can only estimate C_p as a single quantity for one instant in time. This presents a problem because our update laws are dependent on the partial derivatives of $\hat{C}_p$ with respect to λ and β. For example, how can we tell how much $\hat{C}_p$ has changed due to λ and due to β separately? This information is required in determining the update laws in (35) and (36) which provide the bounded convergence of $C_p \rightarrow C_p^*$.

To compute these partial derivatives we require an n+1 number of estimates of $\hat{C}_p$ to determine the partial derivatives for n parameters. In our case, n = 2 for λ and β so n+1=3 estimation data points of $\hat{C}_p$ are required. By holding one of the parameters constant while the other is changed according to its respective update law, we are able to come up with the change in $\hat{C}_p$ due to that parameter. The same thing is done in opposite order. Therefore the partial derivatives of both $\frac{\partial \hat{C}_p}{\partial \lambda}$ and $\frac{\partial \hat{C}_p}{\partial \beta}$ can be computed in k = 3 steps. To clarify this method Figure 4.1 shows a flowchart of the alternating method in which these partial derivatives are computed.

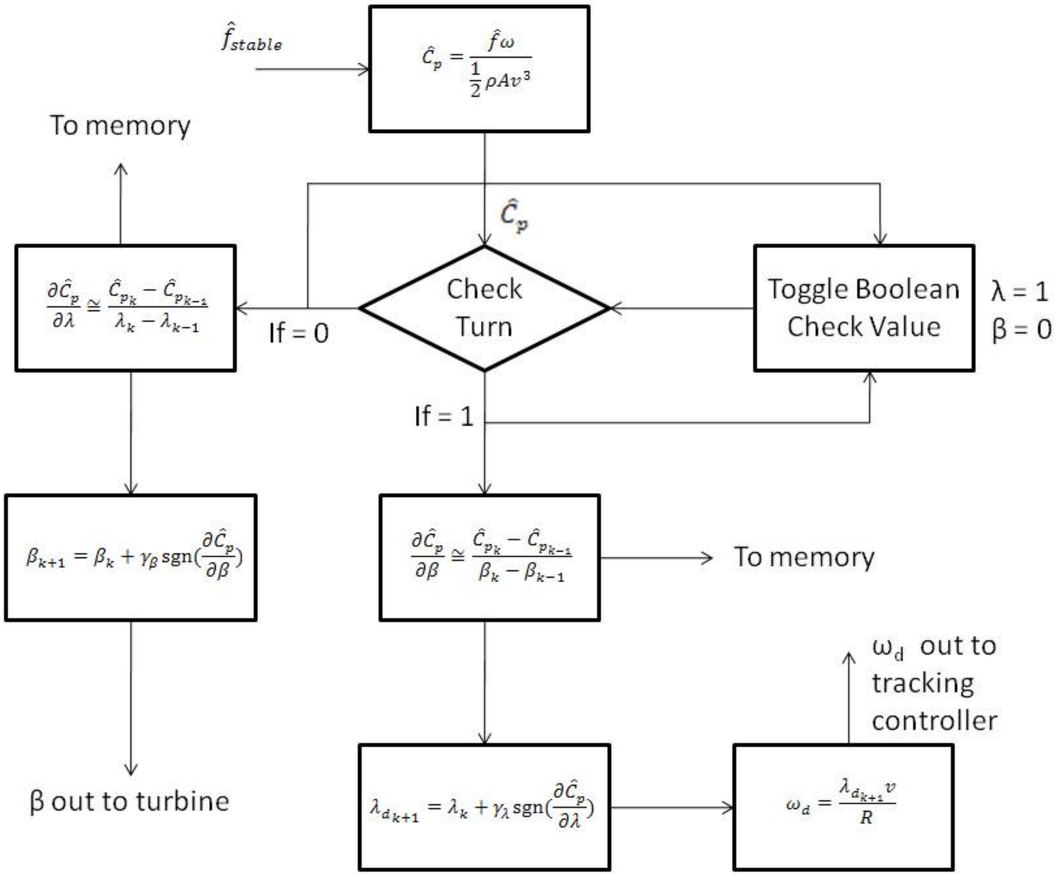
$\hat{f}_{stable}$
$\hat{C}_p = \frac{\hat{f}\omega}{\frac{1}{2}\rho A v^3}$
To memory
$\hat{C}_p$
$\frac{\partial \hat{C}_p}{\partial \lambda} \cong \frac{\hat{C}_{p_k} - \hat{C}_{p_{k-1}}}{\lambda_k - \lambda_{k-1}}$
If = 0
Check Turn
Toggle Boolean Check Value
λ = 1
β = 0
If = 1
$\beta_{k+1} = \beta_k + \gamma_\beta \operatorname{sgn}(\frac{\partial \hat{C}_p}{\partial \beta})$
$\frac{\partial \hat{C}_p}{\partial \beta} \cong \frac{\hat{C}_{p_k} - \hat{C}_{p_{k-1}}}{\beta_k - \beta_{k-1}}$
To memory
ω_d out to tracking controller
β out to turbine
$\lambda_{d_{k+1}} = \lambda_k + \gamma_\lambda \operatorname{sgn}(\frac{\partial \hat{C}_p}{\partial \lambda})$
$\omega_d = \frac{\lambda_{d_{k+1}} v}{R}$

Figure 4.1 Block diagram of the alternating method used for partial derivative computation.

This method begins once the estimate $\hat{f}$ is within a given boundary constrained by the designed parameter ϵ. This parameter determines when the error of the state of the system, given in (12), has been stabilized to a region which is suitable to the application. The aerodynamic torque estimate is given the term $\hat{f}_{stable}$ to denote the value of $\hat{f}$ which will be used in that step of the alternating method. The value of $\hat{f}_{stable}$ is then used to compute the estimate for C_p using equation (25) which yields $\hat{C}_p$.

The next stage is denoted by the rhombus shaped block entitled “Check Turn”. This block checks to see which parameter’s turn it is to be estimated, and which parameter is to be held constant until the subsequent step. If the value of this Boolean variable is equal to one, the estimate of $\hat{C}_p$ is used in the computation of the partial derivative with respect to λ and the Boolean variable is then toggled so that it is equal to 0. On this step the partial derivative of $\frac{\partial \hat{C}_p}{\partial \lambda}$

is calculated according to equation (33) and is stored to be forwarded to the next step. The value of $\frac{\partial \hat{C}_p}{\partial \beta}$, calculated in the previous step (or initialized value for first step), is then used in the update law to find the new set point value for parameter β given in (35). Recall that in this application we are neglecting the blade pitch dynamics so this quantity is updated instantaneously to the wind turbine. If the dynamics were being considered, this value would become the set point for a feedback controller until it was certain that β was stabilized to its desired value.

Alternatively, if the Boolean variable is equal to 0, the estimate of $\hat{C}_p$ is used to determine the partial derivative with respect to β, calculated according to equation (34), and stored for the subsequent step. Similarly the Boolean variable is toggled to 1 so that tip-speed ratio is held constant in the subsequent step. The new set point value of tip-speed ratio is calculated, $\lambda_{d_{k+1}}$ from (36), and then used to compute the new set point, ω_d, according to (37). This value is then provided to the robust controller the new trajectory.

It is important to note that while waiting for stabilization, the trajectory set point ω_d is recalculated for each instance in time to follow the changing wind. The value of $\lambda_{d_{k+1}}$ is held constant in (37) until a new value of $\lambda_{d_{k+1}}$ is calculated two steps later. In this manner, the controller can provide a constant C_p over the stabilization waiting period for time-varying wind conditions. Once the values computed from either leg of this alternating method algorithm are output to the wind turbine, the system must operate at these conditions of λ and β until ω has once again been stabilized within the given tolerance ϵ and the process can begin again.

Error Reduction Techniques

Sources of Estimation Error and Measurement Error

There are two main factors which cause inaccuracy in the proposed discrete alternating method of partial derivative calculation and trajectory generation. The first of these two factors is the estimation error caused by the fact that $\hat{f}$ is not exactly equal to the quantity τ_{aero}. This is known as the estimation error. This error can be modeled as

$$\tau_{aero} = \hat{f} + \zeta, \tag{38}$$

where

$$|\zeta| \leq |c_1|.$$

The value of constant c_1 is the boundary for the estimation error and is determined by the selected stabilization tolerance value ϵ.

Recall earlier in Chapter 4 the inability to calculate the partial derivatives directly. Because we are constrained to using a discrete estimate, an error is introduced during the calculation of $\hat{C}_p$ and furthermore used in the computation of the partial derivatives. As the partial derivatives get smaller near the peak of the C_p power curve, these errors weigh greater on the controller's estimation. The size of error ζ determines the bounds of the convergence of $C_p \rightarrow C_p^*$. If these errors can be reduced, the system will converge with a greater accuracy to C_p^* and be able to sustain operation within that boundary.

The second source of error is the wind velocity measurement instrument. There is a likely probability of white Gaussian noise interfering with the measurement signal coming from the anemometer. Because this modeled system is not actually susceptible to this noise, it is modeled in the control feedback loop. This wind velocity measurement error is modeled as

$$\bar{v}(t) = v(t) + w, \tag{39}$$

where

$$w(t) = \mathrm{N}(0,\sigma)$$

N is the noise signal and the variable σ is the variance of a zero mean white Gaussian noise. This measurement error coupled with, ζ, the estimation error provides the reason for the bounded stability of this control strategy.

In this section, two methods will be developed to reduce the effects of these errors and increase the performance of the controller's ability to operate the turbine closer its optimal power coefficient. The methods will consist of two different strategies. First, a continuous function will be developed which weights the Lyapunov gains of γ_λ and γ_β as a function of $\hat{C}_p$. This will reduce the influence of estimation error near the top of the C_p curve. Second, an array of previous data points is built. Linear curve fitting is then used to develop a more reliable estimation of the partial derivatives in the flat region near the peak of the C_p curve and reduce the influence of wind velocity measurement error.

Error Reduction Using Variable Gain Weighting

Near to the top of the C_p curve, estimation error weighs heavier on the estimation quality of the value of $\hat{C}_p$. The reason for this is variations in λ and β produce smaller changes in the actual value of C_p while the magnitude of possible error is the same. When computing the partial derivatives in this flat region, $\frac{\partial \hat{C}_p}{\partial \lambda}$ and $\frac{\partial \hat{C}_p}{\partial \beta}$ should theoretically approach zero as time approaches infinity. Nearer to the optimal point, the error dominates the $\hat{C}_p$ term of the partial derivative, and the denominator values of $d\lambda$ and $d\beta$ become smaller. The division of the miscalculated $\hat{C}_p$ by these small numbers amplifies the effect of the error which can cause the update law to prescribe large and inaccurate changes according to the $\dot{\lambda}$ and $\dot{\beta}$ update laws.

A method of gain weighting is used to counteract these large and inaccurate disturbances due to estimation error. While not reducing the bounds of the error, ζ, directly, gain weighting does reduce the influence of estimation error on the global behavior of the system and improves the convergence to the optimal region as well as the ability to sustain operation in this region.

A continuous function, shown in Figure 4.2, is designed to weight the gains of γ_λ and γ_β, represented generally by γ in Figure 4.2, as a function of $\hat{C}_p$. The general form chosen for this application is a hyperbolic tangent function.

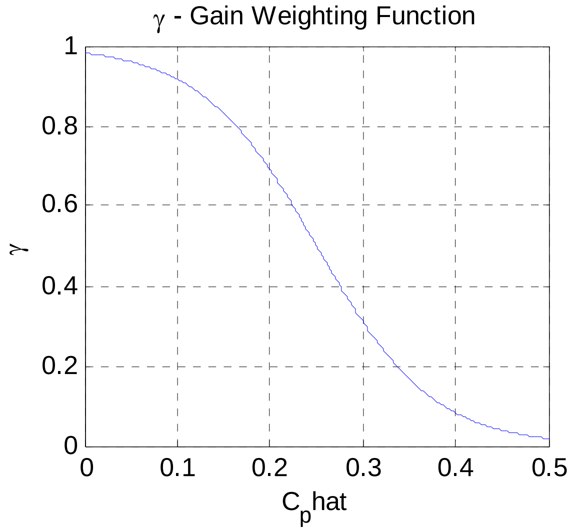

Figure 4.2 Example of a hyperbolic tangent gain weighting function. The function prescribes high gains for low values of $\widehat{C}_p$ and lower gains for values of high $\widehat{C}_p$.

The gains are weighted in the following manner,

$$\gamma = A - B * tanh(C * \hat{C}_p - D), \tag{40}$$

where A, B, C, and D are positive constants determining the shape of the behavior of the weighting function. Figure 4.2 shows the general shape of the gain weighting function.

As shown, gains are larger for smaller values of $\hat{C}_p$ where the partial derivatives are steeper and easier for the finite differences of (33) and (34) to determine. As estimation error holds greater influence for larger values of $\hat{C}_p$, the gain is reduced producing smaller steps in direction specified by the partial derivative computation. If the step is errant, it does not contribute to large changes in λ or β; the estimation of C_p stays relatively near the optimal point.

One disadvantage of using this technique is that the knowledge of the general behavior of the system greatly helps the performance of this error reduction technique when choosing the constants of A,B,C, and D that dictate the shape of this hyperbolic tangent function. It is not however necessary that the behavior is known. If the shoe doesn't exactly fit the foot, the result could be spikes due to amplified miscalculation or suboptimal performance due to over conservativeness.

The good news is that this function can be shaped to handle a generally wide range of operating regions. The parameter C controls the sloped region in the middle of the curve. This gain weighting function can be tuned to become more linear and universal in application. An example can be seen below in Figure 4.3. This shape still achieves the same global result while providing a more universal fit for a wider range of C_p behaviors for various types of turbines.

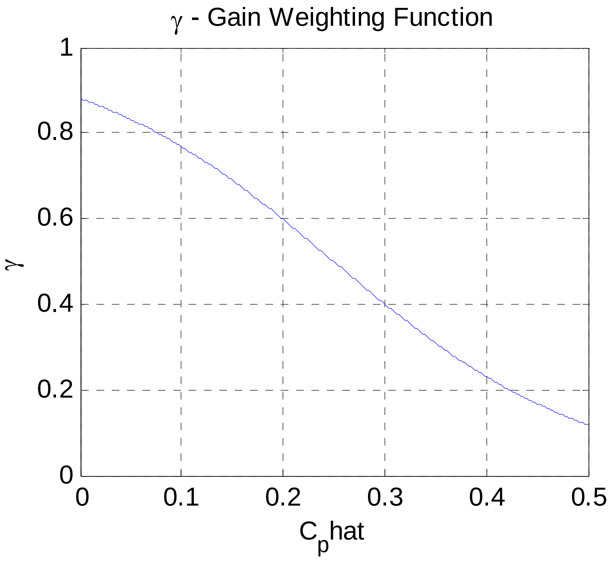

Figure 4.3 Example of a hyperbolic tangent gain weighting function. The constants have been tuned to give it a more universal shape when system behavior isn't well known.

It is also possible to use other weighting functions instead of the hyperbolic tangent. One other possible weighting function is

$$A + \frac{B(\hat{C}_p + D)}{C + |\hat{C}_p + D|}.$$

This function may be easier for implementing on controllers that aren't equipped to calculate the hyperbolic tangent. This function is also very similar in behavior to the hyperbolic tangent and can be shaped by tuning the parameters A, B, C, and D.

Error Reduction Using Linear Curve Fitting

Another technique can be used to benefit the convergence behavior which reduces the effects of both estimation and measurement error. In equations (33) and (34), two points are used to estimate the partial derivatives of C_p with respect to λ or β by means of finite difference. This is sufficient when the slope of the C_p curve is steep because the error does not have as great an influence on this gradient estimate. When the differences become smaller in the flat region near the top of the C_p curve, the influence of the estimation and measurement error play a greater role. One approach to reduce the effect of these errors is to look at more than just the current and previous data point.

This error reduction approach takes a look at several previous data points and then computes the equation of a line using linear curve fitting. Two n x 2 arrays are built containing the last 2n data points. The first array holds n coordinates of $\hat{C}_p$ and its corresponding value of λ when $\hat{C}_p$ was calculated. The second array holds n coordinates of $\hat{C}_p$ and its corresponding value of β. The 1st column contains data points for previous values of $\hat{C}_p$ and the second column contains either the values of λ or β depending on the array.

Recall in the Alternating Method of Chapter 4 that between each step of the alternating method, waiting is required until $\hat{f}$ is determined useful and then $\hat{C}_p$ is calculated. Typically, at this point the finite difference, (33) or (34), is taken to estimate the partial derivatives. This method uses more than just the previous k and k-1 data steps. It employs an n-1 number of previous samples and the current kth sample to fit a linear equation whose slope is then used to determine the partial derivatives. The size of n chosen for the wind turbine application is given later in Chapter 5, Table 5.4.

Figure 4.4 shows a diagram of the manner in which these two arrays are constructed. They both follow a first-in first-out convention, cycling the newest data values into the array and the oldest out.

Beta Array

$\hat{C}_{p_k}$	β_k
$\hat{C}_{p_{k-2}}$	β_{k-2}
$\hat{C}_{p_{k-4}}$	β_{k-4}
$\vdots$	$\vdots$
$\hat{C}_{p_{k-(2n-1)}}$	$\beta_{k-(2n-1)}$

Lambda Array

$\hat{C}_{p_{k-1}}$	λ_{k-1}
$\hat{C}_{p_{k-3}}$	λ_{k-3}
$\hat{C}_{p_{k-5}}$	λ_{k-5}
$\vdots$	$\vdots$
$\hat{C}_{p_{k-2n}}$	λ_{k-2n}

Figure 4.4 Diagram of array construction for the linear curve fitting error reduction method.

The k-2 increment is due to the alternation of set-point determination of λ and β due to the alternating method. Each array holds n data points. The MATLAB 'polyfit' function is then used to compute a linear curve fit with λ as the independent variable and $\hat{C}_p$ as the dependent variable.

This linear equation is then differentiated using ‘polyder’ and evaluated to determine the partial derivative with respect to either λ or β depending on the leg of the alternating method.

This method reduces the effect of estimation error by smoothing errant steps due to miscalculation. Using this method, the influence of each data point carries the weight of 1/n so an errant step due to estimation error will carry less weight. In the finite difference method of (33) and (34), each data point carries the weight of 1/2.

Wind velocity measurement error is also reduced by this technique because of the nature of the measurement noise which rides on the wind velocity signal. It is presumed that the measurement noise is a zero mean white noise. Because linear curve fitting is the similar to averaging, as the number of points n in the array increases, the less the measurement noise will affect the estimation.

CHAPTER 5 - Simulation of the Wind Turbine Control Method

A simulation is conducted of a wind turbine system and proposed control methods. The wind turbine dynamics, presented in Chapter 2, are modeled to simulate the behavior of an actual wind turbine. The controller developed in Chapter 3 is used to regulate the system state ω of the system about set-point, ω_d, as well as provide the Lyapunov-based extremum seeking controller with an estimate of the unknown aerodynamic torque. The Lyapunov-based controller, developed in Chapter 4, is then employed to compute a set-point trajectory which ensures the bounded convergence of the power capture coefficient close to its optimal value. The presented error reduction techniques are also tested to analyze their improvement on the system's performance in the presence of estimation error and measurement noise. The parameters used in the simulation will be presented as well as the results and a discussion of the findings.

The wind turbine system and controllers are modeled using the Simulink tool in the MATLAB software package. This simulation also employs blocks from the Simulink Wind Turbine Blockset 3.0 [33]. The blocks in the simulation can be broken up into the following main six subsystems: Wind Generator, Wind Turbine Rotor, Drive Train Dynamics, Robust Estimator Controller, Measurement Noise Generator, and the Lyapunov-Based Extremum Seeking Controller. The simulation block diagram can be seen below in Figure 5.1. The wind generator and the wind turbine rotor blocks come from the wind turbine blockset. For further information on their function beyond what is presented in this thesis, please refer to its documentation [33].

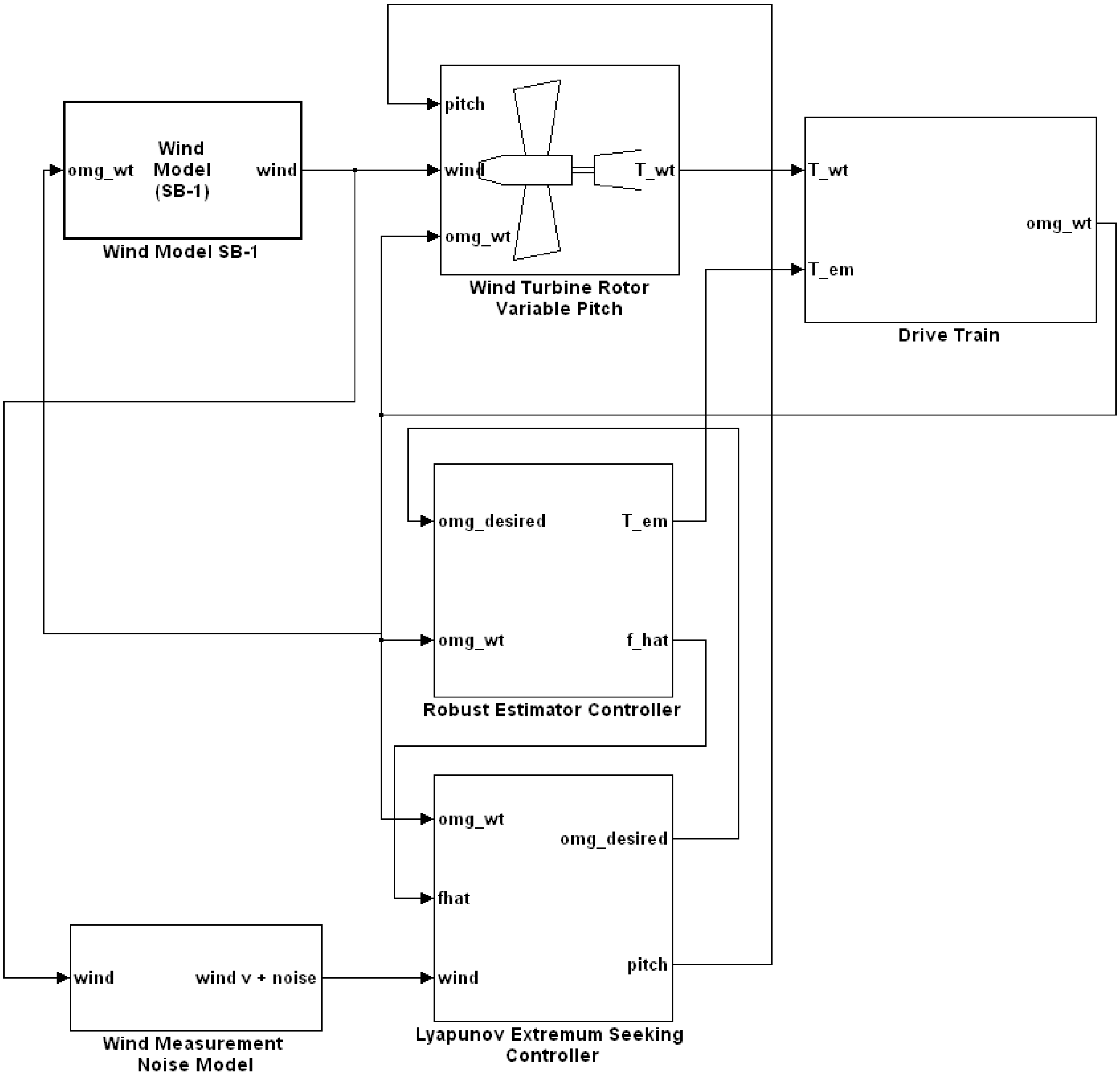

Figure 5.1 Simulink Simulation Diagram of Subsystems

The wind generation block uses a Kaimal statistical model to generate a realistic wind signal with some specified mean value and turbulence percentage. The wind turbine rotor block emulates the wind capture behavior. Based on the parameters of wind speed, blade pitch, and angular velocity, this block computes the captured aerodynamic torque based on a realistically modeled C_p function. The drive train implements the differential equation given in (5). The robust estimator is the control of Chapter 3. The noise generation block generates zero mean white Gaussian noise which is not part of the wind signal to the system but only in the feedback loop. Finally the Lyapunov-based control block implements the Lyapunov-based extremum seeking method of Chapter 4. The error reduction methods are computed within this block as well.

A wind turbine is modeled with the parameters listed in Table 5.1 below. The parameters were selected to model a large commercial scale wind turbine on the order of 1.0MW or greater.

Table 5.1 Simulation Model Constants

Parameter Variable	**Parameter Value**	**Units**	**Description**
J	100,000	kg·m^2	Total Moment of Inertia
R	40	m	Rotor Blade Radius
ρ	1.25	kg/m^3	Density of Air
C_D	1.0	kg·m^2/sec	Damping Coefficient
λ^*	7.8	--	Optimal Tip-Speed Ratio Value
β^*	-1	deg	Optimal Blade Pitch Value
C_p^*	0.472	--	Optimal Coefficient of Performance Value

The performance coefficient curve used to simulate the wind turbine power capture is taken from the data accompanying the wind turbine blockset. Two plots of the data are shown below in Figures 5.2 and 5.3. The first, figure 5.2, shows the entire data set for $-90 \leq \beta \leq 90$ and $0 \leq \lambda \leq 20$. The values of C_p are largely negative for most of the data points in this plot. The second graph, figure 5.3, shows a zoomed in view of the area of C_p which is positive. This region is important because it contains the information of C_p for the values of λ and β in the region of operation. During simulation, C_p for values of λ and β which fall between points on the data table is found through linear interpolation.

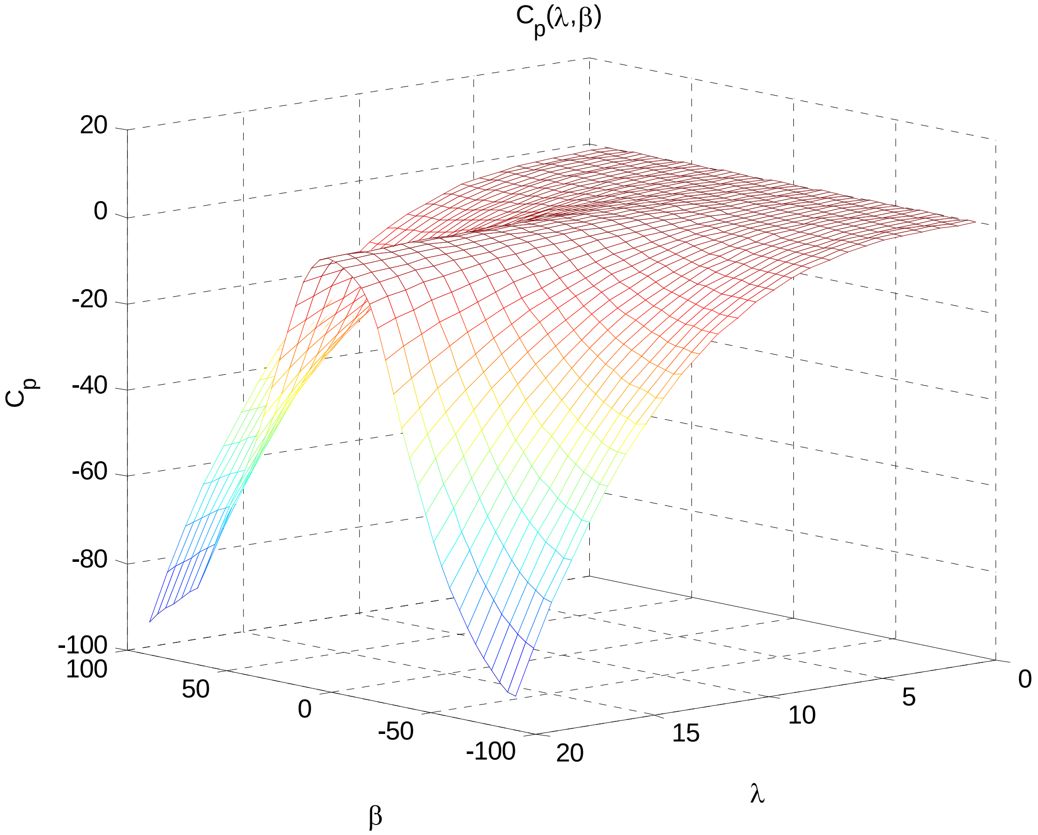

Figure 5.2 Graph of *C_p* curve for wide range of *λ* and *β* values.

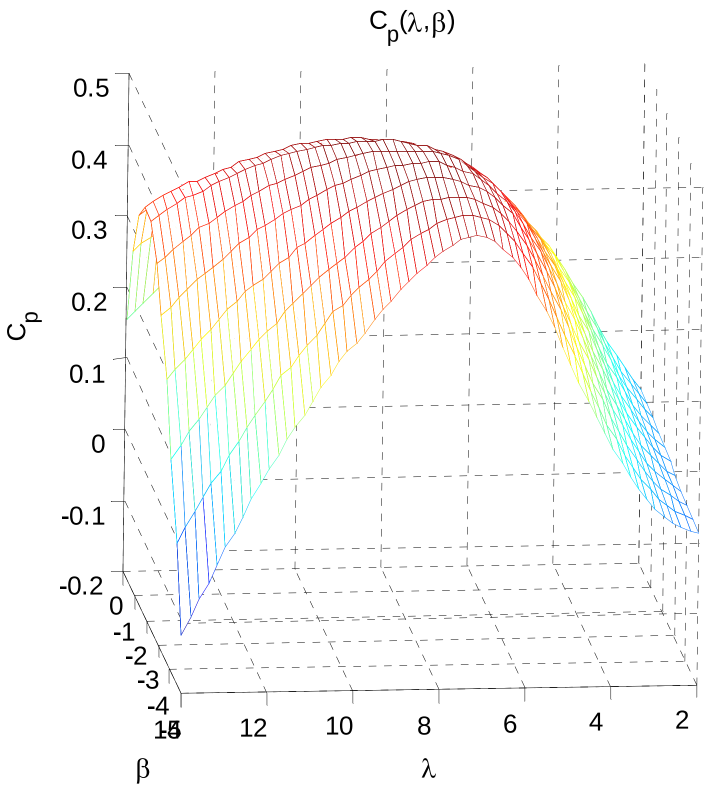

Figure 5.3 Graph of C_p curve for operating region of λ and β values. The operating region is $0 \leq \lambda \leq 14$ and $-5^o \leq \beta \leq 5^o$

The following sections will display the results gathered by the simulation of the control strategy on the wind capture system model. The simulations will include an analysis of the performance of the Lyapunov-based control scheme alone in the first test. The second test will show the performance of the system with the addition of the error reduction techniques developed at the end of Chapter 4.

Test 1: Simulation Using Lyapunov-Based Controller

The first simulation of the turbine will include the robust estimator controller and the Lyapunov-based extremum seeking strategy without the addition of the two error reduction techniques. This simulation is meant to show the performance of the proposed control strategy alone. The wind turbine is modeled and simulated using values from the tables below. Table 5.2 shows the values of the robust controller gains and extremum seeking controller gains as well as other designable parameters of the Lyapunov-based controller such as generator saturation and error tolerance value ϵ.

Table 5.2 Test 1 Controller Gain Values

Parameter	**Value**	**Description**
k_s	10,000,000	Robust Controller Gain
β_c	20	Robust Controller Gain
α	7	Robust Controller Gain
U_{max}	$25*10^5$	Generator Saturation (Torque)
U_{min}	0	Generator Saturation (Torque)
ϵ	0.001	Regulation Tolerance (ω- ω_d)
γ_λ	0.25	Extremum Seeking Gain
γ_β	0.5	Extremum Seeking Gain

The parameters k_s, β_c, and α are gains for the robust controller. The values of U_{max} and U_{min} are saturation values which correspond to the maximum and minimum torque values that the generator can provide. The maximum value is determined by taking the rated generator power and dividing by the maximum shaft angular velocity. The minimum has been set at zero providing a bound on the control so the generator is always extracting power from the wind and never draws power from the grid. The value ϵ is set to 0.001 which means that once the error signal (ω - ω_d) is within the asymptotic bounds of $\pm\epsilon$, the value of $\hat{f}$ is acceptable to use for estimation of the aerodynamic torque which is used in turn to calculate $\hat{C}_p$. The values of γ_λ and γ_β have been selected so that the convergence of $C_p \to C_p^*$ is relatively fast without producing extremely large spikes once within the region of C_p^*.

Table 5.3 below shows the parameters involving the simulation. These parameters include the initial conditions of the system, the wind signal, parameters for noise generation, and simulation run time.

Table 5.3 Test 1 and 2 Simulation Parameters and Initial Conditions

Parameter	Value	Description
ω_0	0.01	Shaft Velocity Initial Condition (rad/s)
β_0	8	Blade Pitch Initial Condition (deg)
$v(t)$ avg	10	Average Wind Velocity (m/s)
$v(t)$ turb	12	Wind Turbulence (%)
$w(t)$	N(0,0.05)	White Measurement Noise Normal(mean,var)
T	250	Simulation Run Time (sec)

Some of the figures in this chapter will show a zoomed out and then a zoomed in view of each plot. This is due to the long simulation times of the system. The two views will give the reader an understanding of the system behavior on a global level over time and also the behavior up close. These figures correspond to the first test run with no error reduction techniques. The first figure shows the tracking of $\omega \rightarrow \omega_d$.

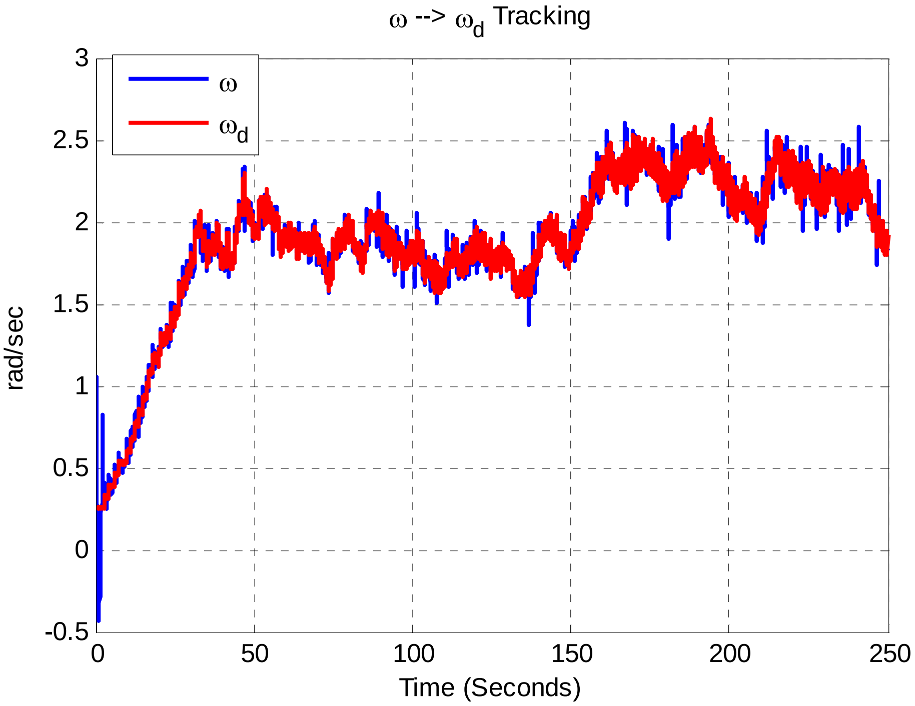

Figure 5.4 Test 1: Tracking of the shaft angular velocity to the desired trajectory. The red shows the desired path and the blue shows the actual wind turbine angular velocity ω.

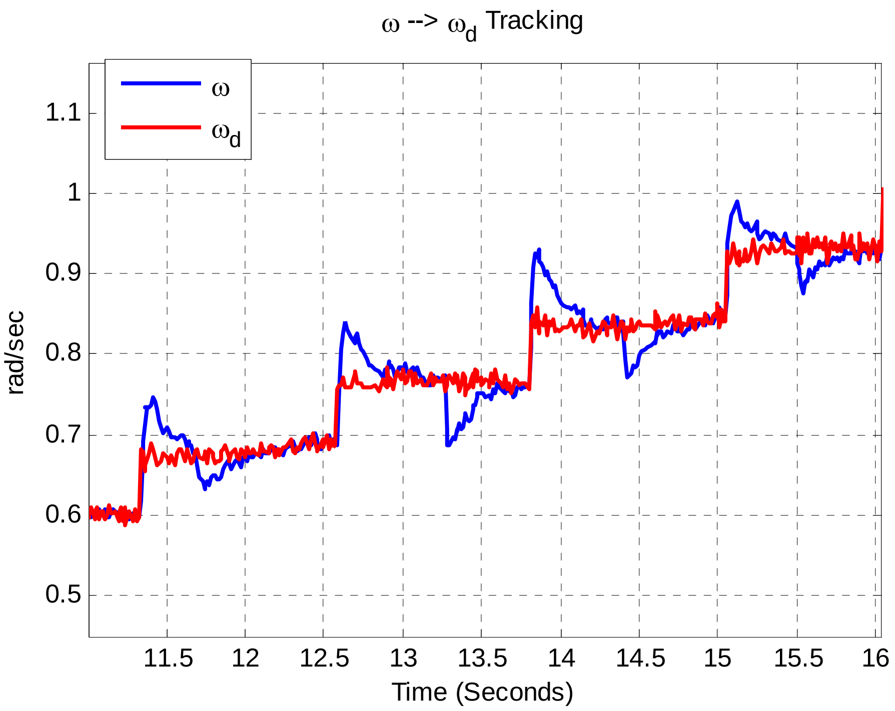

Figure 5.5 Test 1: Tracking of the shaft angular velocity to the desired trajectory. The red shows the desired path and the blue shows the actual wind turbine angular velocity *ω*. (Zoom)

As shown in Figures 5.4 and 5.5, the robust controller regulates the turbine so that the system reaches stability within the specified bounds. Once within a certain error tolerance, the extremum seeking controller selects a new set point for the regulator. Notice how there are two spikes for each flat region in Figure 5.5. This shows the alternating selection of a new ω_d set point, holding β constant, and then a new β set-point while holding ω_d constant.

Figures 5.6 and 5.7 show the tracking of λ about the λ_d set point value.

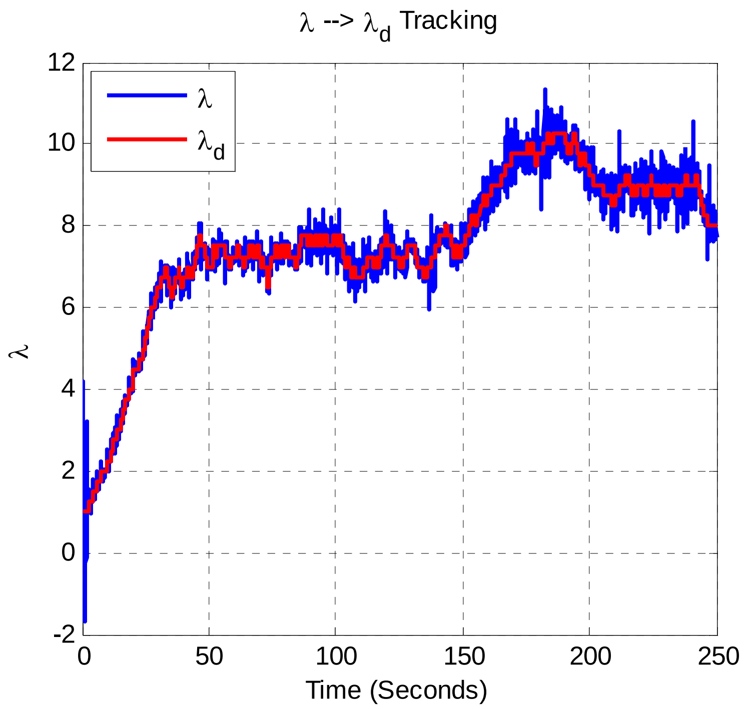

Figure 5.6 Test 1: Tracking of the tip-speed ratio to the desired trajectory. The red shows the desired path and the blue shows the actual wind turbine tip-speed ratio *λ*.

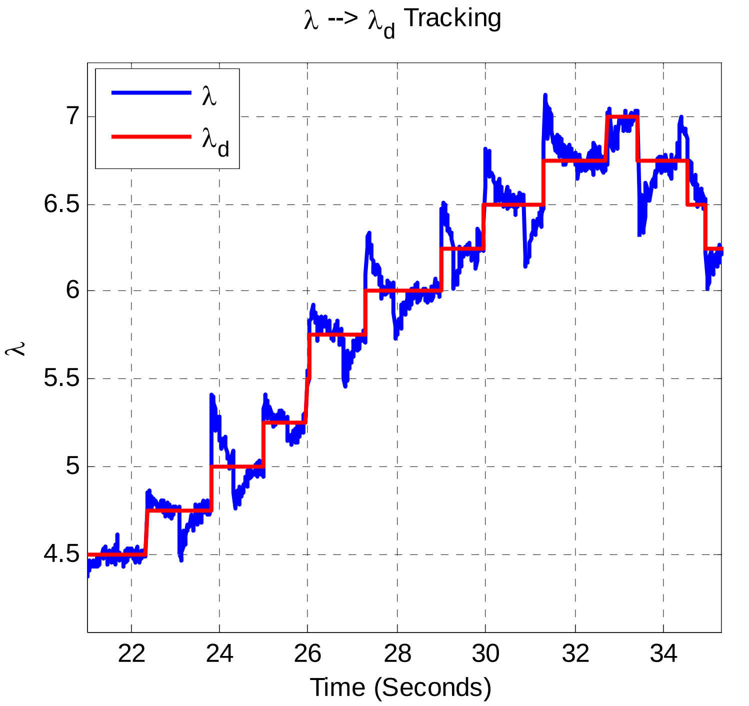

Figure 5.7 Test 1: Tracking of the tip-speed ratio to the desired trajectory. The red shows the desired path and the blue shows the actual wind turbine tip-speed ratio *λ*. (Zoom)

Figures 5.6 and 5.7 demonstrate that regulating the shaft angular velocity to its desired set point value also regulates the tip-speed ratio to its set point value. The desired value of λ_d is held constant until it is regulated, within tolerance, for both steps of the alternating method. This also verifies that once regulation has occurred, the values of lambda used in the calculation by the extremum seeking controller are accurate.

Next the accuracy of estimation is analyzed. Recall that once ω is regulated within some error tolerance ϵ of ω_d then the estimate of τ_{aero}, $\hat{f}$, is considered to be suitable for use. In the following plots, Figures 5.8 and 5.9, the accuracy of the estimate is displayed by plotting τ_{aero}, in red, and $\hat{f}$, in blue.

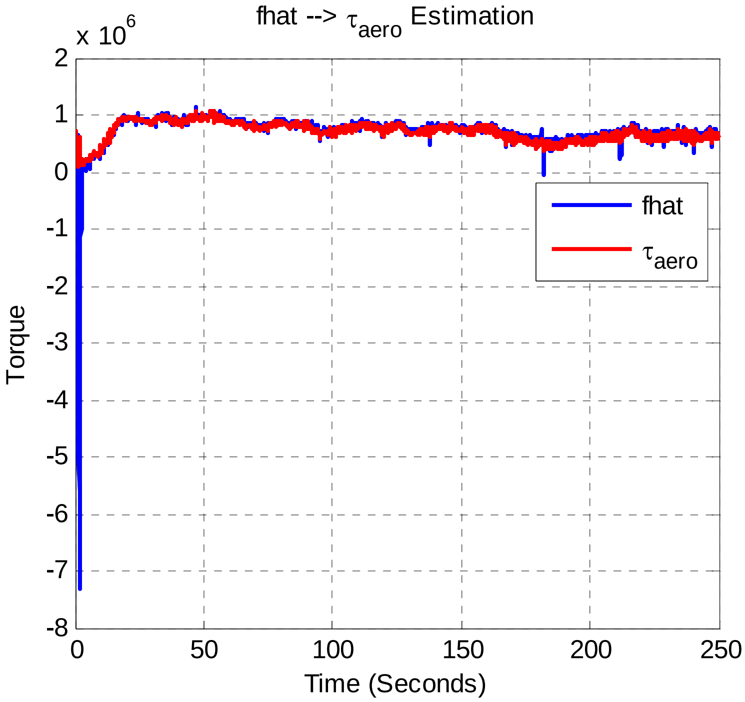

Figure 5.8 Test 1: This plot shows the accuracy of $\hat{f}$ in its ability to estimate the unknown aerodynamic torque, τ_{aero}.

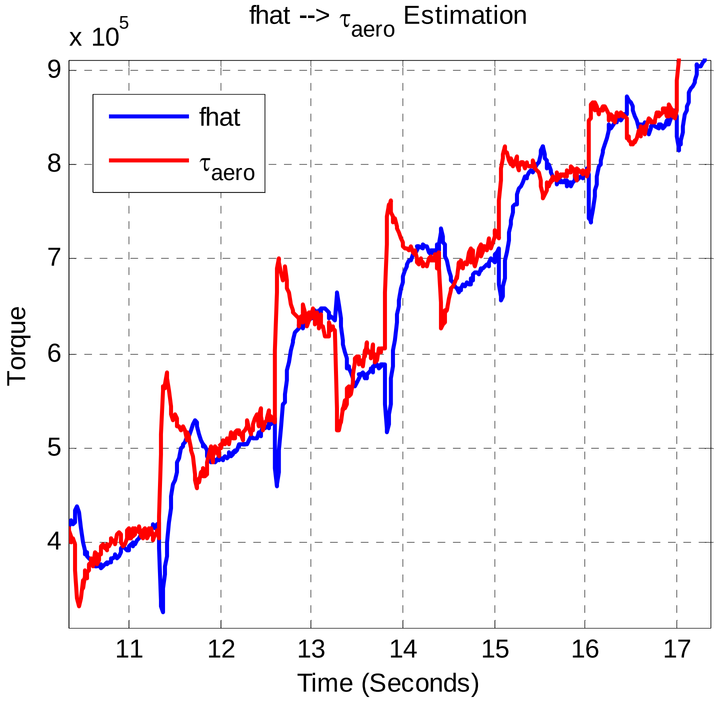

Figure 5.9 Test 1: This plot shows the accuracy of $\hat{f}$ in its ability to estimate the unknown aerodynamic torque, τ_{aero}. (Zoom)

Figure 5.8 shows the global tracking of the estimate to the actual value of aerodynamic torque. When zoomed in, Figure 5.9, the tracking does exhibit some error as predicted. Recall that this error was denoted by ζ. It is also important to note that the values of $\hat{f}$ being used for the estimation of C_p are being collected once the stability of state ω has been regulated. For each set point trajectory this is where the estimation error is the smallest.

Let us now examine the accuracy of the calculation of $\hat{C}_p$. Recall that this calculation involves the use of the parameter $\hat{f}$ and also the measurement of wind velocity, both of which contain error. In the following figures, the estimation $\hat{C}_p$ is shown in blue and the actual value of C_p is shown in red.

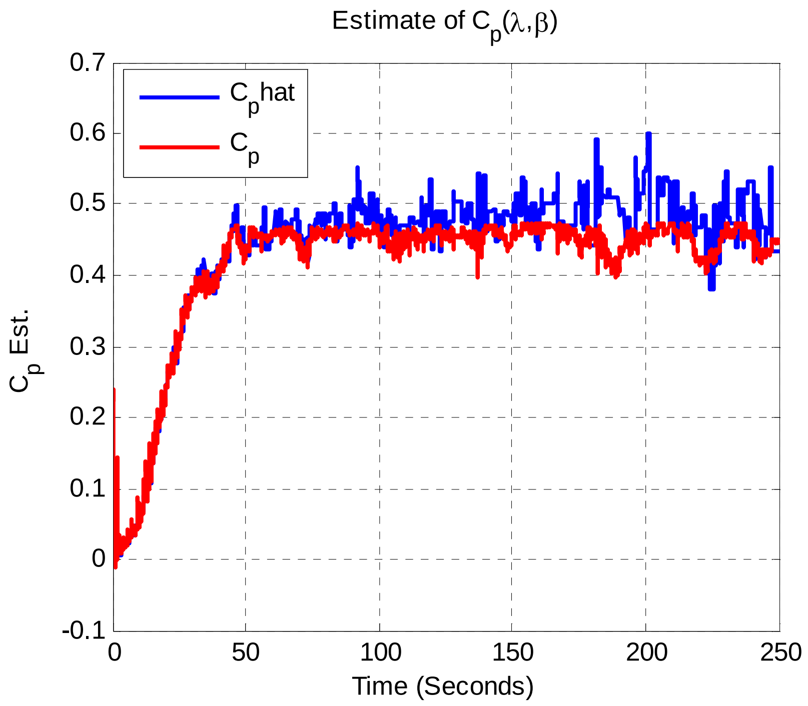

Figure 5.10 Test 1: This plot shows a comparison of the estimation of $\widehat{C}_p$to the actual system value C_p.

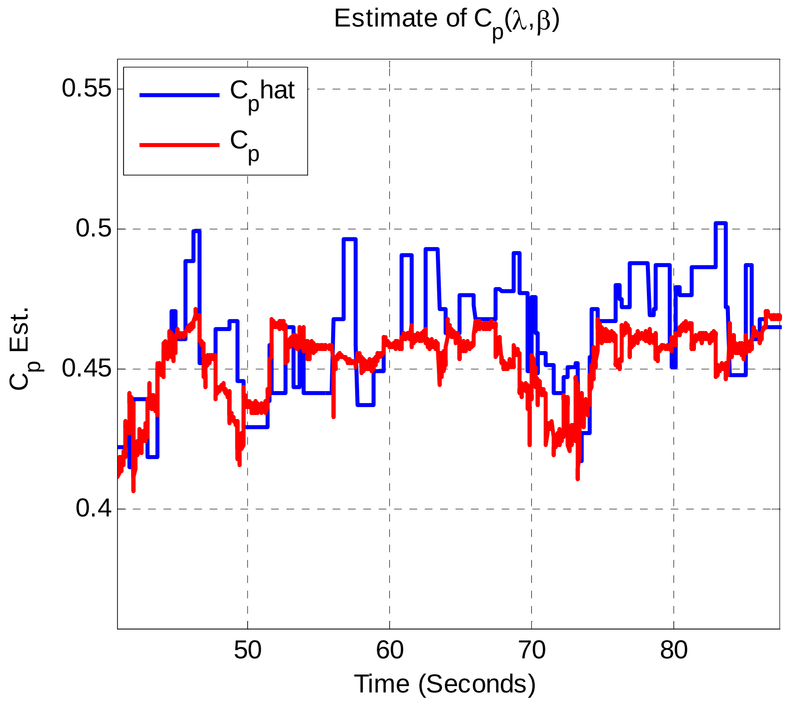

Figure 5.11 Test 1: This plot shows a comparison of estimation of $\widehat{C}_p$to the actual system value C_p. (Zoom)

Based on Figure 5.11, the local estimation of the performance coefficient is not too precise. The reasons for this are the estimation error and the measurement error in the wind velocity. Figure 5.10 shows, however, that the global behavior of the estimation has the correct trend. Later figures which display the convergence C_p to the optimal value, show that the performance coefficient is maximized in spite of this error in calculation.

Below is a figure of the measurement noise modeled into the system. The measurement noise signal is a zero mean normally distributed random number. This noise signal is added to the wind velocity signal and applied to the controller. Therefore, the noise carries the same units as the wind velocity in m/s. [Note: The noise is only added to the wind velocity measurement.]

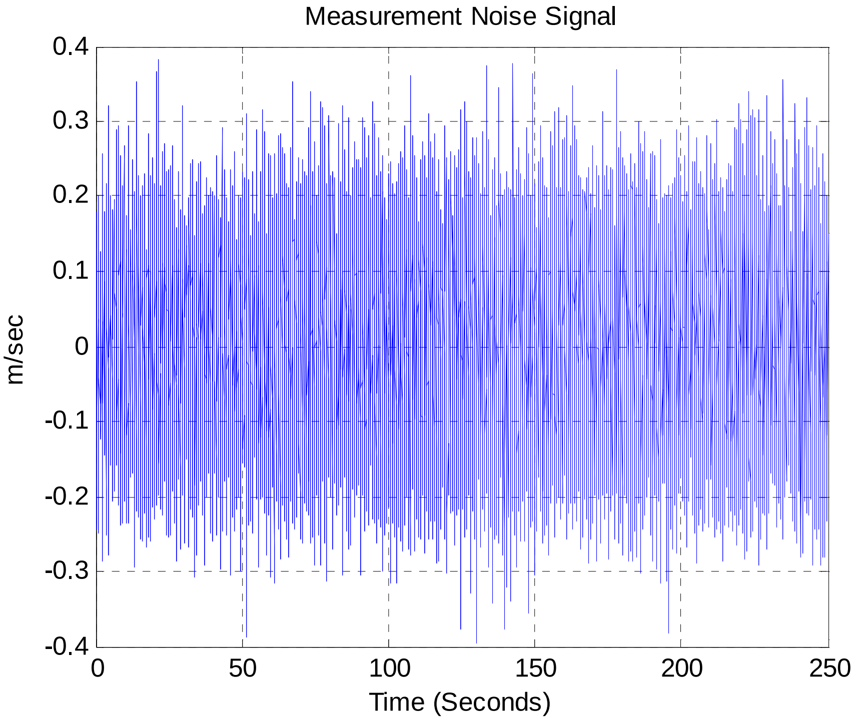

Figure 5.12 Test 1: This plot shows the generated measurement noise which is added to the wind velocity signal and applied to the outer control feedback loop.

Figures 5.13 - 5.18 show the generated wind velocity input, the convergence of parameters λ, β, and C_p to their optimal values, and the captured wind power. Figure 5.13 is the wind signal applied to the wind turbine generated by the Kaimal wind function from the blockset. Figure 5.14 shows the desired value of λ, represented by blue, and its bounded convergence behavior about the optimal tip-speed ratio value, λ^*, shown in red. Figure 5.15 shows the similar bounded convergence behavior of β about β^*, same color scheme. Figure 5.16 is a plot of the coefficient of performance. The red, constant line is indicative of the optimal value of C_p^* and the blue line is the actual value of C_p. It should be noted that this value is not known to the controller nor is it practically measurable but is valuable to analyze the performance of the controller. This measurement has been extracted from the simulation to show the performance of the system only. Finally Figures 5.17 and 5.18 show a comparison of the power captured by the turbine to its possible power capture ability normalized by the optimal value of $C_p^* = 0.4725$.

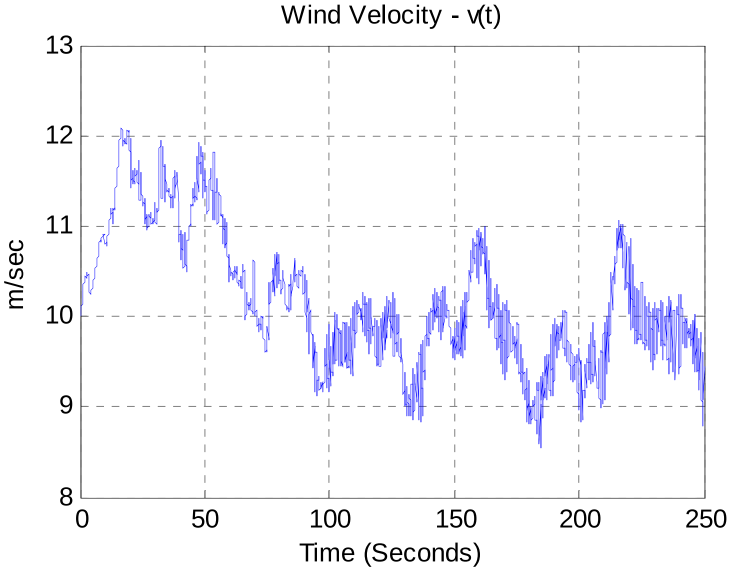

Figure 5.13 Test 1: Plot of wind velocity measurement over time. This figure does not include the modeled measurement error.

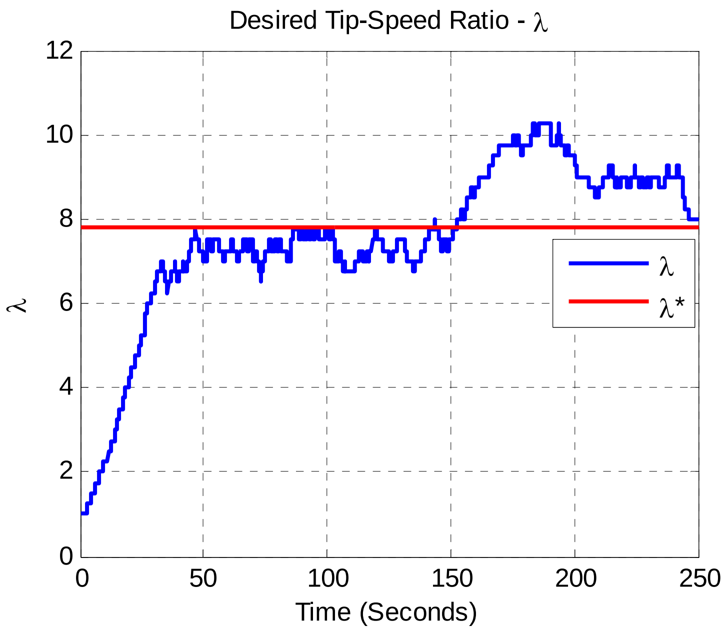

Figure 5.14 Test 1: Plot of trajectory generated by the extremum seeking controller for λ_d in blue and λ^* plotted in red.

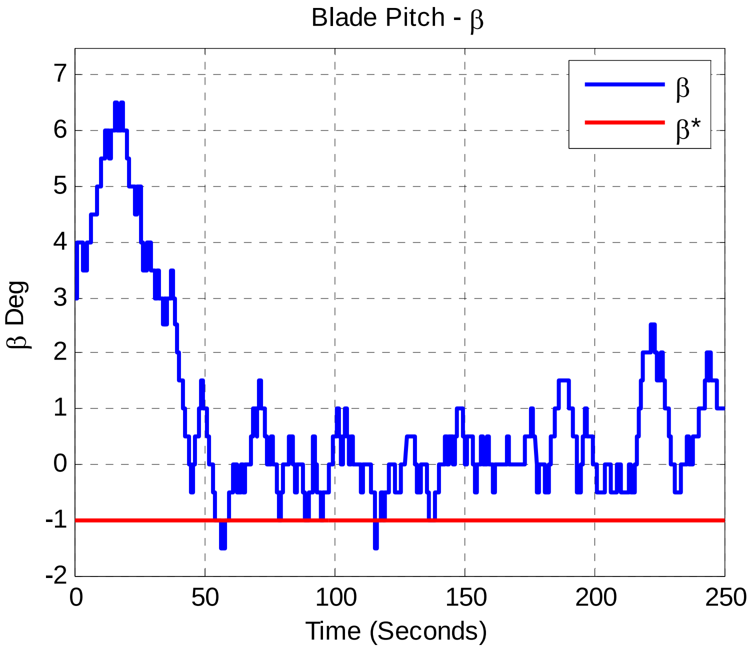

Figure 5.15 Test 1: Plot of trajectory generated by the extremum seeking controller for β_d in blue and β^* plotted in red.

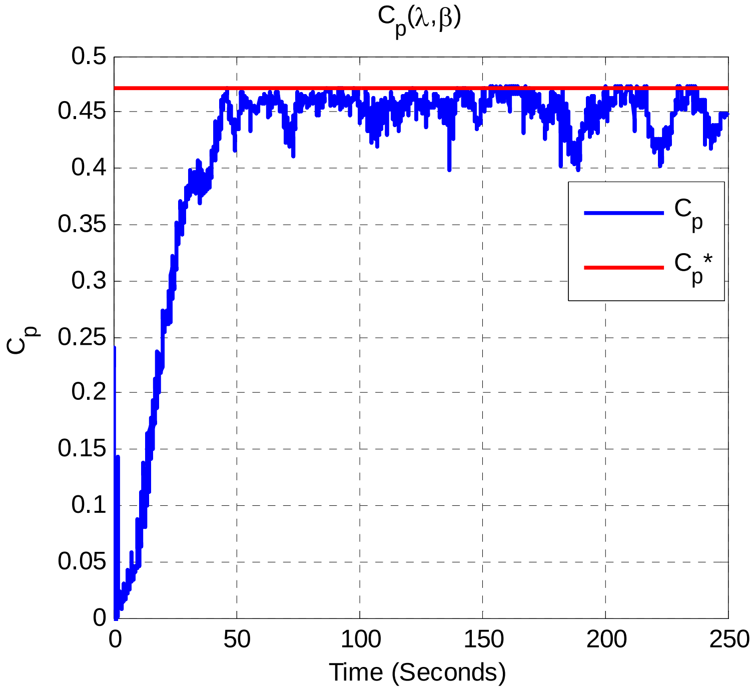

Figure 5.16 Test 1: Plot of the actual system coefficient of performance, C_p, as it converges toward the optimal value C_p^*

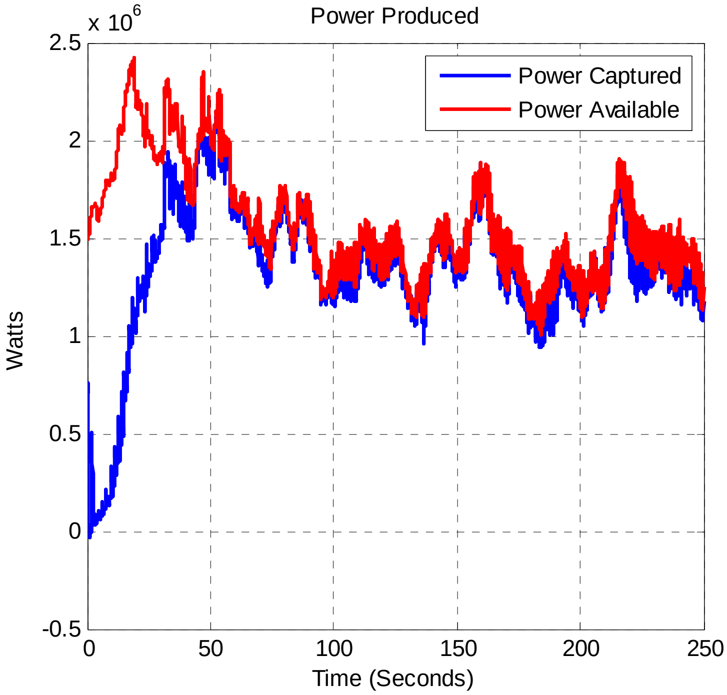

Figure 5.17 Test 1: This plot shows the mechanical power extracted from the wind in blue compared to the possible power that the turbine would capture if operating at the peak power coefficient.

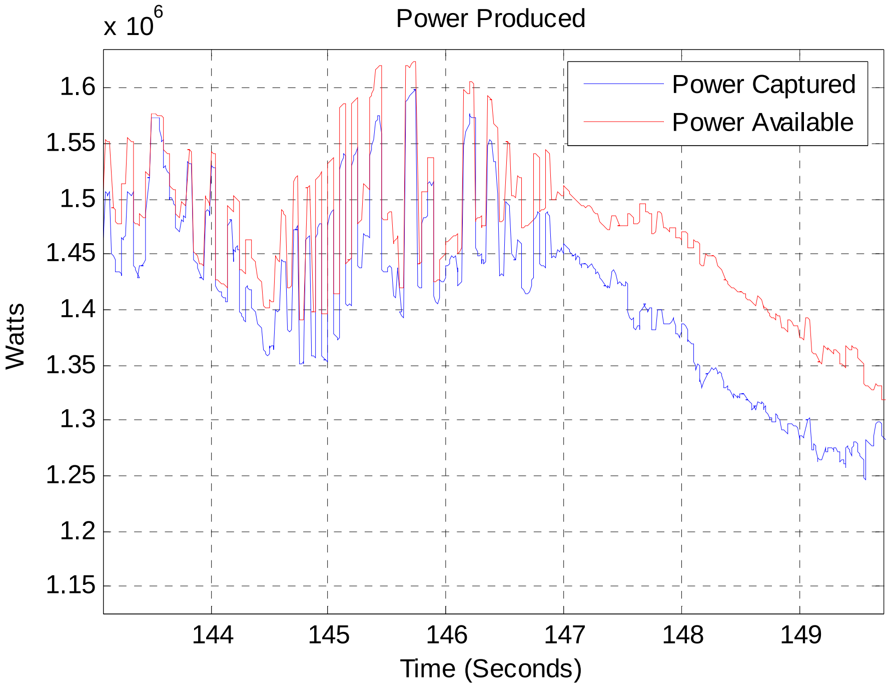

Figure 5.18 Test 1: This plot shows the mechanical power extracted from the wind in blue compared to the possible power that the turbine would capture if operating at the peak power coefficient. (Zoom)

Figures 5.13 – 5.18 show the results of the first simulation. As shown in Figure 5.16, as time goes on C_p converges to near the area of its optimal value of $C_p^* = 0.472$. The controller performs well in the scope of the control goal which is the bounded maximization of C_p. The coefficient of performance is capable of reaching its optimal value and stays within a bounded region. Figure 5.16 also demonstrates that the power coefficient maintains its position near the optimal value over time, seldom dropping below $C_p = 0.4$.

Notice that C_p is increasing toward its optimal point until it reaches it and then varies away from C_p^*. Due to the flat nature of the $\mathrm{C_p}$ surface in this region, the effects of estimation and measurement error come into play. This causes the desired values of tip speed ratio and blade pitch to take errant steps due to miscalculation. The behavior of the system can also be affected by the tuning of the robust estimator controller gains and also adjusting the values of γ_λ and γ_β. Typically choosing larger values for either γ results in a faster initial convergence to C_p^* but causes larger jumps away from it while trying to maintain that operation. Conversely, choosing smaller values for γ reduces the size of the jumps away from the optimal point but results in very long converge times. In some cases where γ is selected very small, the Lyapunov controller may not ever produce a desired trajectory which converges to the value of C_p^* due to the 'flat' nature of the top of the C_p curve.

Figure 5.17 shows the power captured by the turbine in blue. The red shows the optimal power capture corresponding to the maximum value of C_p which is 0.472. The power captured tracks the trajectory of the optimal power but only slightly less. Figure 5.18 shows a zoomed in view to get a more small scale view of the power capture. While the power captured is shifted down from the optimal, the tracking dynamic of the system is very fast. Figures 5.16, and 5.17 and 5.18 demonstrate that the control goal of increasing power capture by maximizing C_p has been achieved. The results of this simulation indicate that this control approach is a successful technique for the maximization of Region II power capture.

Test 2: Simulation Using Lyapunov-Based Controller and Error Reduction

In this section a simulation will be run similar to Test 1 but will employ the use of the two methods of error reduction developed at the end of Chapter 4. The goal of using the error reduction techniques is to avoid large jumps away from the optimal point. This test demonstrates how the error reductions help maintain operation near the optimal point of C_p^* to greater enhance the effectiveness of the Lyapunov-based strategy. The system will still be modeled according to the values listed in Table 5.1. The initial conditions, wind, noise and run time used in this simulation will also be the same as Test 1 and listed above in Table 5.3. Below, Table 5.4 shows the gains and design parameters used in this second test.

Table 5.4 Test 2 Controller Gain Values and Error Reduction Values

Parameter	Value	Description
k_s	3,000,000	Robust Controller Gain
β_c	2	Robust Controller Gain
α	5	Robust Controller Gain
U_{max}	$25*10^5$	Generator Saturation (Torque)
U_{min}	0	Generator Saturation (Torque)
ϵ	0.001	Regulation Tolerance (ω- ω_d)
A	0.75	Hyperbloic Tangent Vert Shift
B	0.75	Hyperbloic Tangent Height
C	12	Hyperbloic Tangent Shaping
D	4.5	Hyperbloic Tangent Horiz shift
n	40	Array Size (# of data points)

The following figures represent data collected from Test 2 which included the error reduction techniques of gain weighting, for extremum seeking gains γ_λ and γ_β, and linear curve fitting from collecting previous data points. The first figures show the tracking of state variable ω to ω_d.

Figures 5.19 and 5.20 show that the tracking of the angular velocity is achieved for each trajectory set point. Figure 5.19 demonstrates the tracking of the state in a global sense through time. Figure 5.20 shows more clearly how the regulation occurs for each individual set point. We can see here that the controller is functioning correctly by regulating the angular velocity to the set point. Once within the designed tolerance, the extremum seeking controller computes the next set point for regulation.

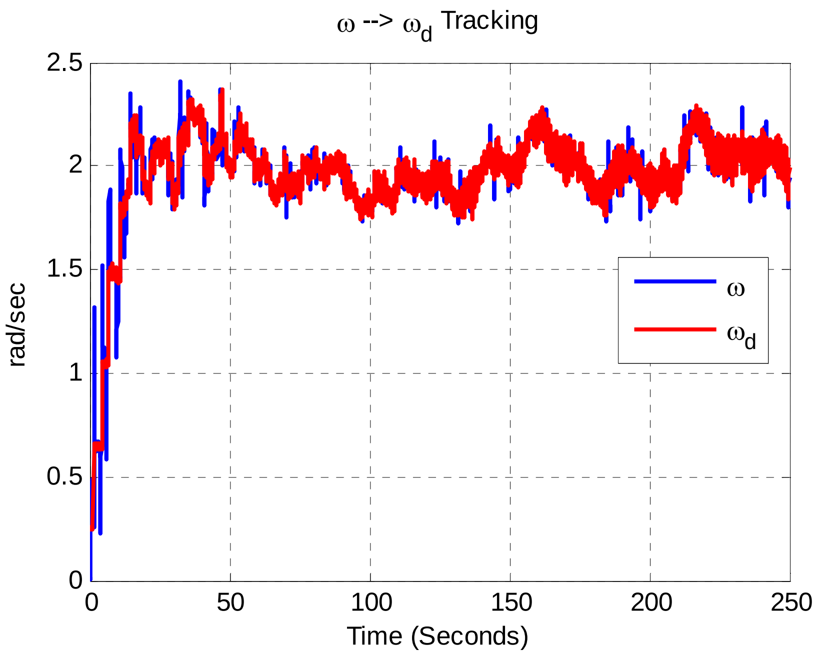

Figure 5.19 Test 2: Tracking of the shaft angular velocity to the desired trajectory. The red shows the desired path and the blue shows the actual wind turbine angular velocity *ω*.

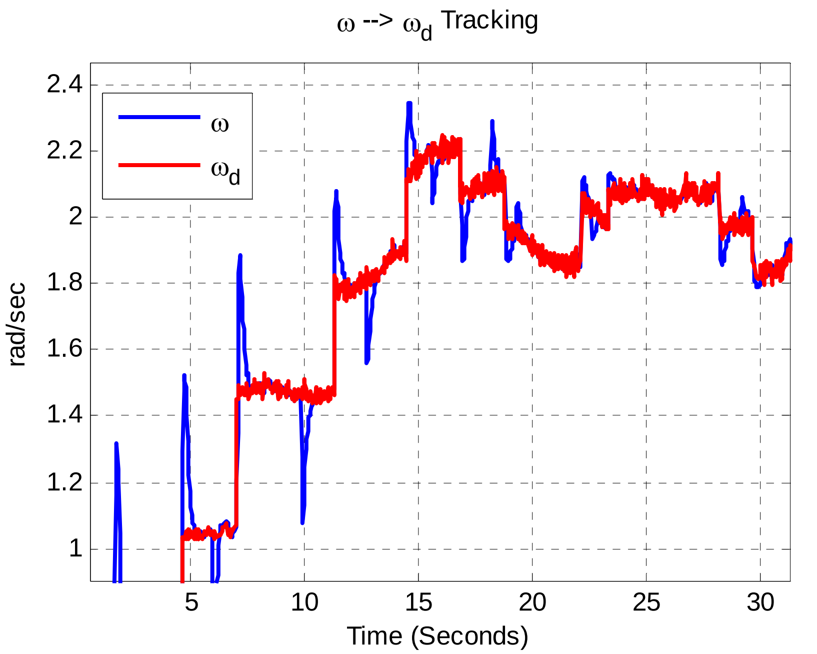

Figure 5.20 Test 2: Tracking of the shaft angular velocity to the desired trajectory. The red shows the desired path and the blue shows the actual wind turbine angular velocity *ω*. (Zoom)

Figures 5.21 and 5.22, below, show the regulation of the tip-speed ratio about the desired set point value.

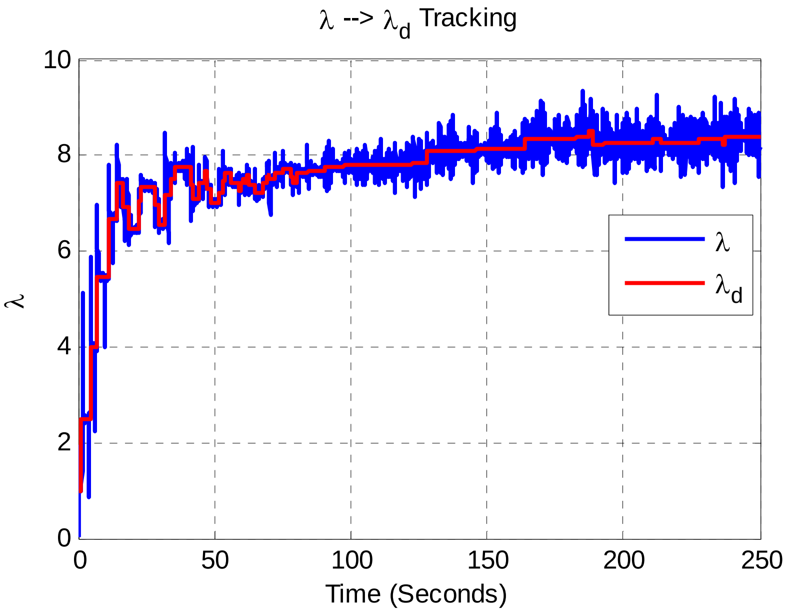

Figure 5.21 Test 2: Tracking of the tip-speed ratio to the desired trajectory. The red shows the desired path and the blue shows the actual wind turbine tip-speed ratio *λ*.

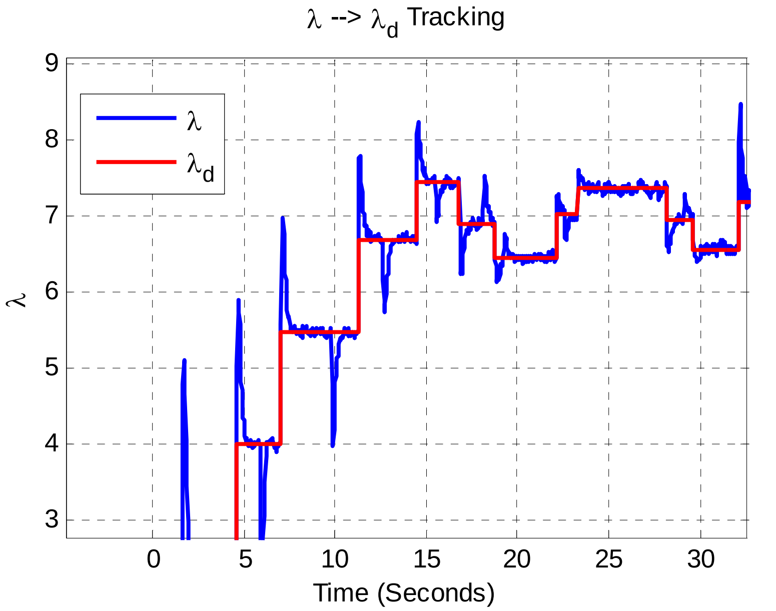

Figure 5.22 Test 2: Tracking of the tip-speed ratio to the desired trajectory. The red shows the desired path and the blue shows the actual wind turbine tip-speed ratio *λ*. (Zoom)

Figure 5.21 shows the global sense of the tip-speed ratio tracking. As shown, the tracking toward the beginning is much better than near the end. As $\lambda \rightarrow \lambda^*$, the wind turbine reaches the flatter region of the C_p curve. In this region the measurement error plays a larger factor in the quality of the signal. Recall that the set point for the angular velocity is calculated as a function of wind speed. Although the angular velocity is being regulated well, the tip-speed ratio experiences the effects of the measurement noise. Although the measurement error is present, Figure 5.22 shows that on a smaller scale, the tip-speed is being regulated but the convergence is very noisy.

Next let us look at the quality of the estimation of τ_{aero} for this second test. Figure 5.23 and 5.24 show that the estimation of the aerodynamic torque is accurate on global scale and at the level of each set point. According to the development of the robust estimator controller, the value of $\hat{f}$ will approach the value of τ_{aero} as $t \rightarrow \infty$, but it will contain some error, ζ, because we only let t approach some finite value and then decide a new set point for the trajectory. These figures support that the theoretical development is correct. Recall also that the value of $\hat{f}$ used for the calculation of C_p is taken at the point of smallest error just before the trajectory change.

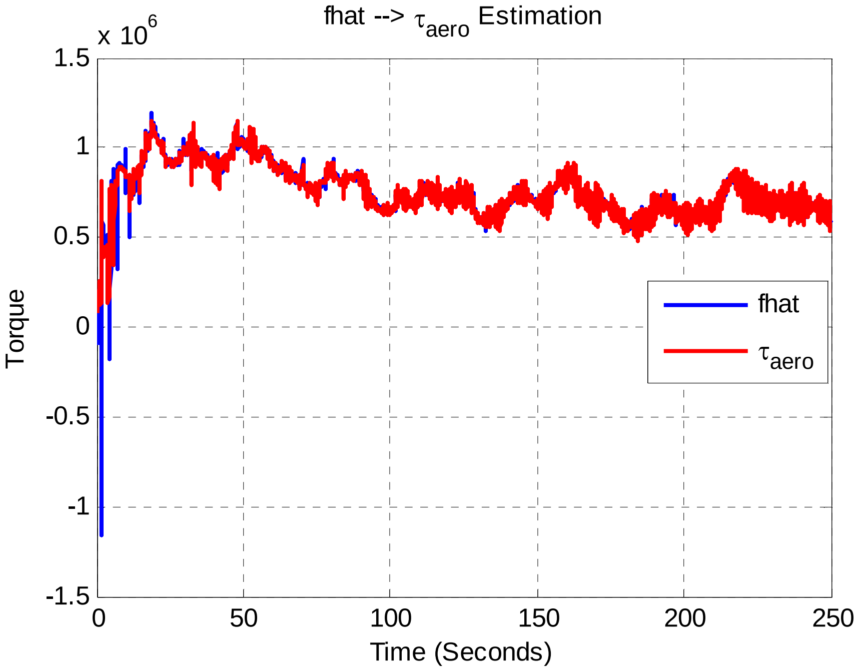

Figure 5.23 Test 2: This plot shows the accuracy of $\hat{f}$ in its ability to estimate the unknown aerodynamic torque, τ_{aero}.

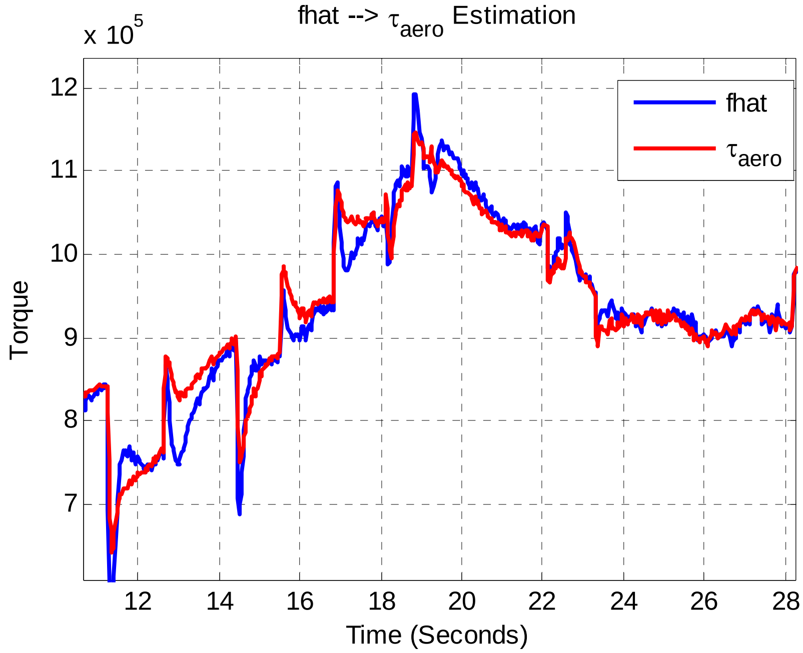

Figure 5.24 Test 2: This plot shows the accuracy of $\hat{f}$ in its ability to estimate the unknown aerodynamic torque, τ_{aero}. (Zoom)

Now let us look at the closeness of the estimate of $\hat{C}_p$ compared to C_p. $\hat{C}_p$is a function of the estimation of aerodynamic torque, containing the error ζ, and the wind velocity, containing the measurement error w. Because of the error contained in the estimation of C_p, it is not 100% accurate. Figure 5.26 shows this in the local convergence of $\hat{C}_p$ to C_p. Figure 5.25 compares the global result of the estimation to the exact value.

Test 2 employs the use of the error reduction techniques to minimize the effects that the measurement and estimation error have on the maximization of the quantity C_p. We can see the results of these methods in Figure 5.25. Although error in the estimation of $\hat{C}_p$ is large on both the positive and the negative side, the actual behavior of C_p is not affected adversely. The improvement in Figure 5.25, compared to the Figure 5.10 of Test 1, is due largely to the error reduction technique of linear curve fitting. Recall that linear curve fitting behaves as averaging. The red line of actual C_p trends to the average value of the $\hat{C}_p$ estimation, shown in blue.

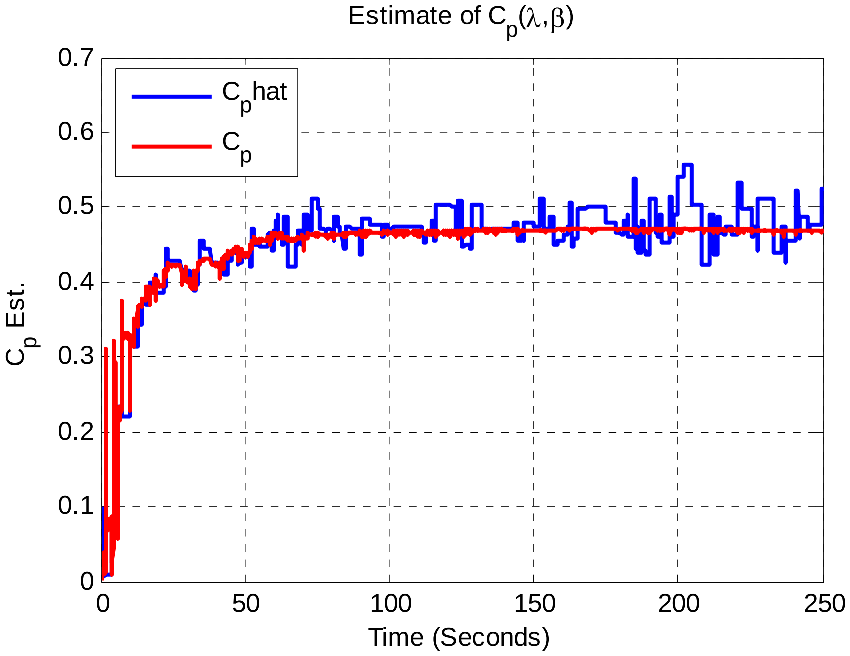

Figure 5.25 Test 2: This plot shows a comparison of estimation $\widehat{C}_p$to the actual system value C_p.

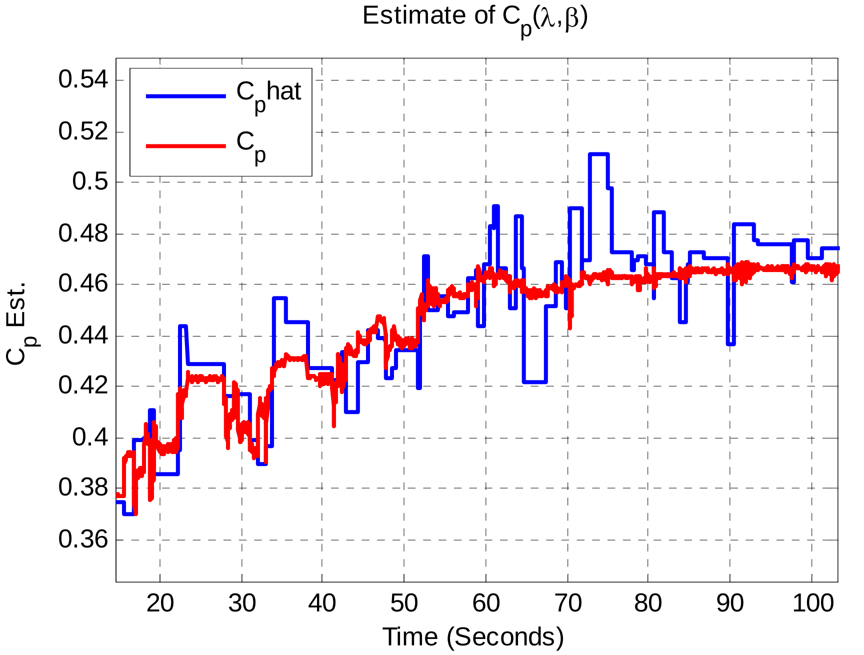

Figure 5.26 Test 2: This plot shows a comparison of estimation $\widehat{C}_p$to the actual system value C_p. (Zoom)

Finally the measurement noise is shown in Figure 5.27. Similar to Test 1, the measurement noise is added to the wind velocity signal in the feedback control loop and carries the same units of wind velocity in m/s.

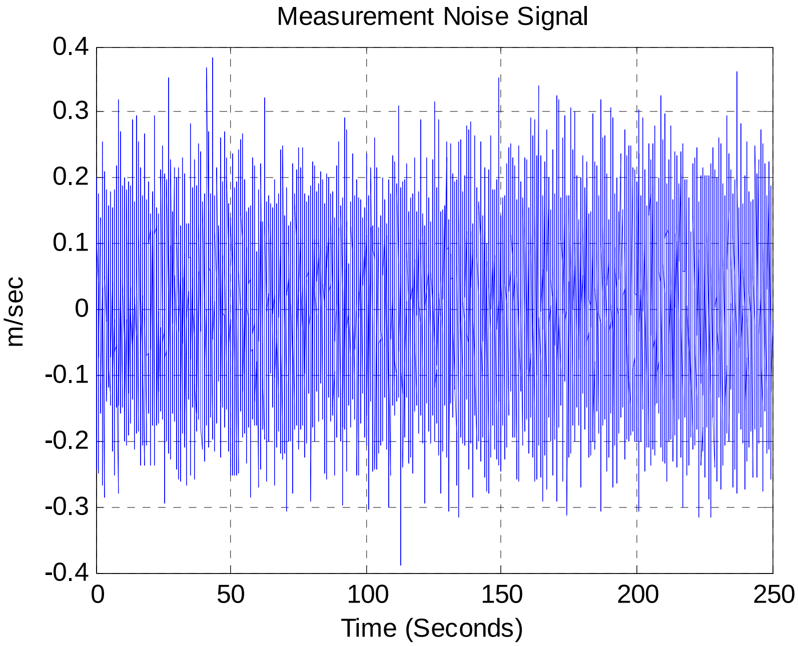

Figure 5.27 Test 2: This plot shows the generated measurement noise which is added to the wind velocity signal and applied to the outer control feedback loop.

Figure 5.28 is the wind signal for the system. Figure 5.29 shows the desired trajectory of λ in blue and λ^* in red. Figure 5.30 shows the generated trajectory for state variable β and the optimal value of β^*. The performance coefficient is shown in Figure 5.31 along with its optimal value denoted by the red line. Finally, the power captured by the turbine is displayed in Figures 5.32 and 5.33.

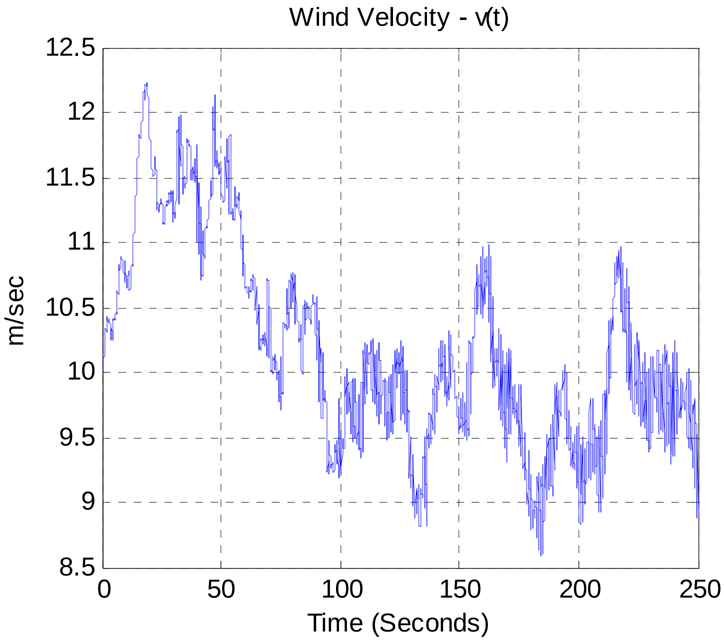

Figure 5.28 Test 2: Plot of wind velocity measurement over time. This figure does not include the modeled measurement error.

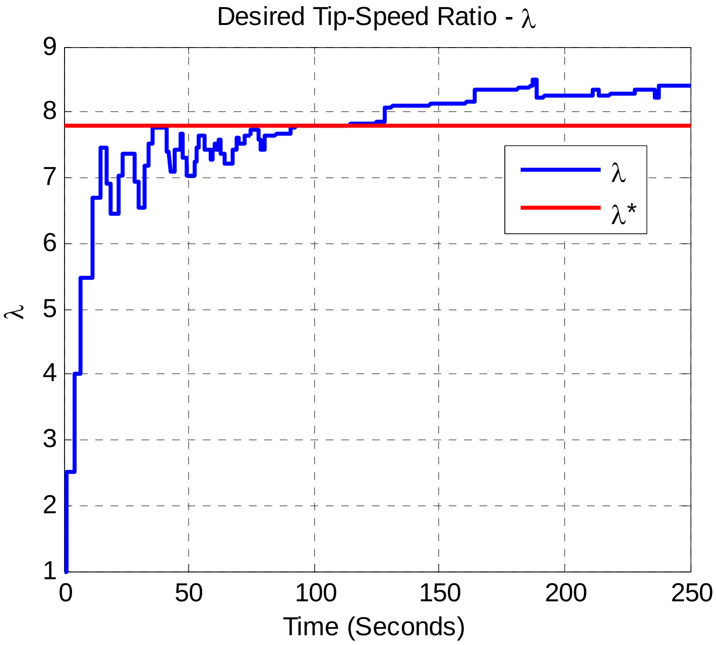

Figure 5.29 Test 2: Plot of trajectory generated by the extremum seeking controller for λ_d in blue and λ^* plotted in red.

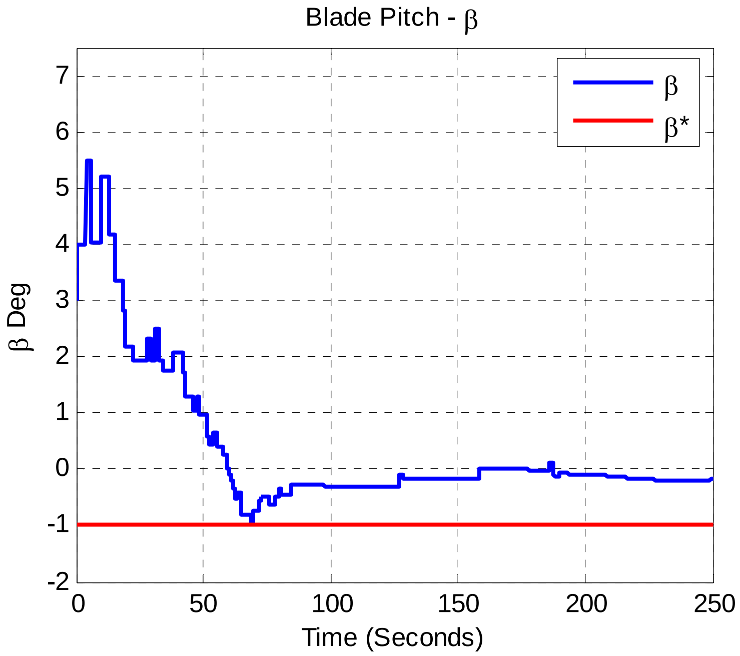

Figure 5.30 Test 2: Plot of trajectory generated by the extremum seeking controller for β_d in blue and β^* plotted in red.

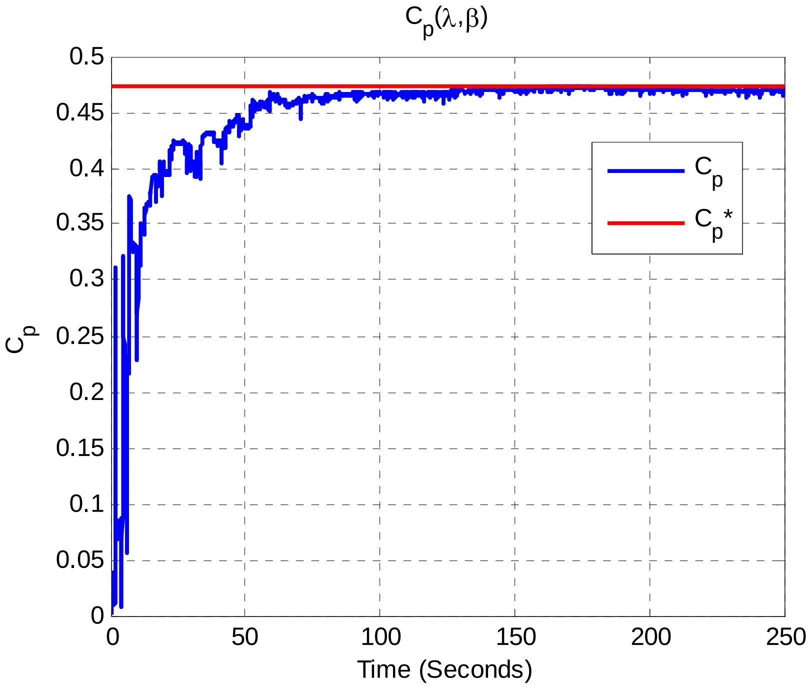

Figure 5.31 Test 2: Plot of the actual system coefficient of performance, C_p, as it converges toward the optimal value C_p^*

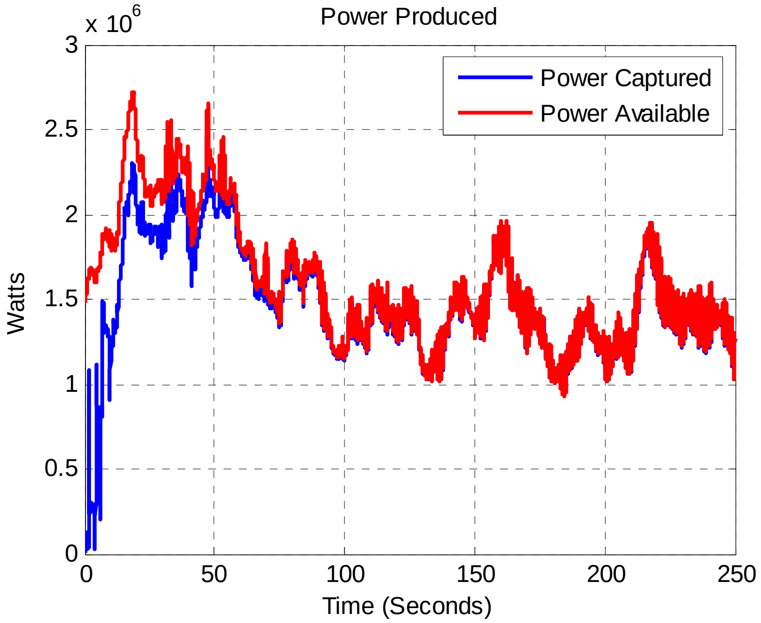

Figure 5.32 Test 2: This plot shows the mechanical power extracted from the wind in blue compared to the possible power that the turbine would capture if operating at the peak power coefficient.

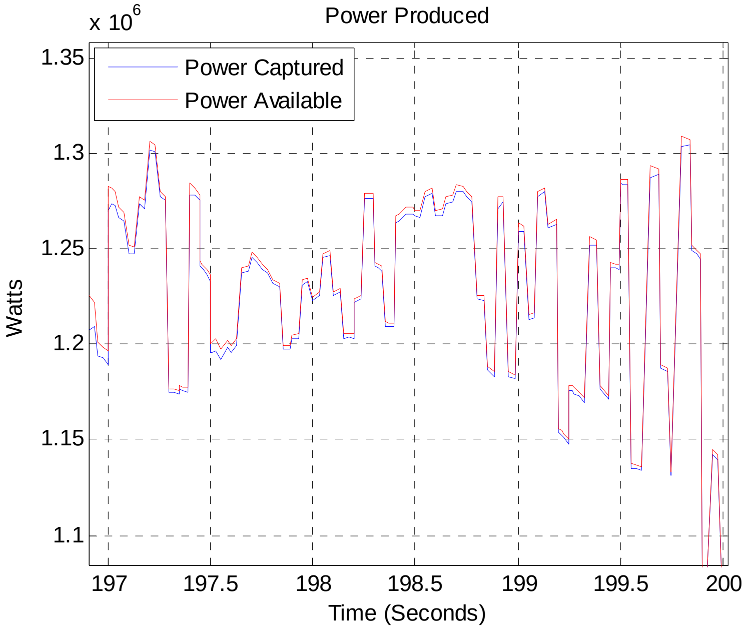

Figure 5.33 Test 2: This plot shows the mechanical power extracted from the wind in blue compared to the possible power that the turbine would capture if operating at the peak power coefficient. (Zoom)

As shown in the figures above the use of error reduction in addition to the proposed control method has met the control goal of maximizing C_p within a region near to C_p^*. The use of the two error reduction techniques has also enabled the controller to maintain the operation of the turbine at this near optimal level without large jumps away from the optimal value. The values of λ and β are also close to the values of λ^* and β^* and maintain their close proximity to those values. The step size between the desired trajectory values diminishes as the estimate of C_p increases providing a smoother tracking trajectory due to gain weighting.

One other result of this second test is the reduction of mechanical vibration in the system. Notice in Figures 5.29 and 5.30 that once the state variables, λ and β, become close to their optimal operating values, the controller prescribes smaller changes in their trajectory. In comparison with the Test 1, these trajectories are much smoother which induces less vibration on the system. This can also be seen in Figure 5.31 as the value of C_p stays much more constant with less bouncing. The power capture of Figure 5.32 is almost ideally matched to the power capture that this turbine is able to achieve. Test 1 showed that the control goal was met through the implementation of the robust controller Lyapunov extremum seeking controller alone. By employing the error reduction methods, we have improved the performance of the original control scheme to reduce the addition of vibration and mechanical stress as well as improve the energy capture performance to be near optimal.

In closing, the following notes should be made about the simulation of this wind turbine and control system. For this large-scale wind turbine, we are able to show C_p converge from an initial value near zero to a region very near to the optimal value within approximately 59 seconds. The initial conditions are intentionally chosen far away from λ^* and β^* to demonstrate the performance of the controller. In a practical application, the turbine would most likely be initialized very close to its optimal operating region. This would greatly shorten convergence times and may alter the design of controller gains.

CHAPTER 6 - Conclusions and Future Work

A robust estimator is used to provide trajectory tracking control of a system in the presence of an unknown nonlinear disturbance. This robust identifier controller also provides an approximation of the unknown quantity which is used in the calculation of an estimate of the power coefficient. A Lyapunov-based extremum seeking control is then used to generate a desired trajectory which maximizes the unknown performance coefficient. The wind turbine system is modeled to emulate a large commercial-scale turbine with a realistic power coefficient curve and considering measurement noise. This control method has been theoretically developed and simulated to demonstrate its performance in a realistic wind condition. A version of the controller will be presented at the ASME Dynamic Systems Control Conference this coming fall, [34].

The performance of this control method in simulation suggests that it is a viable solution to maximizing wind turbine Region II power capture. The next step for this control scheme is to apply it to an actual test wind turbine, such as CART at NREL, and actual data could be collected. It is my belief that this turbine would perform well in an actual environment based on all of the considerations made in the system model as well as the performance of the controller in simulation.

Several considerations have been made in the controller to model realistic parameters such as generator reaction torque saturation. Other considerations which could be modeled for the future are maximum torque allowances for other mechanical parts of the system. One thing that could also be updated with the model is the inclusion of a gear box which is present in most turbines in practice. In this model the gear box has been omitted for a direct drive system.

The generator dynamics also were not considered. This control law calculates the specified torque required to regulate the system. Future work would contain a model for the generator used to actuate this desired torque. A common generator type used for variable speed wind turbines is a doubly-fed induction generator (DFIG). This type of generator requires a complex model and could be added to the scope of this research. It would make more sense to begin with the model of a DC generator (direct drive or gearbox) and work towards more complex models. It is important to begin simply with the addition of a new dynamic system, especially when it is cascaded with another dynamic system. Another control law for the generator torque following would need to be developed to ensure that the prescribed torque from the extremum seeking controller was being achieved by the generator.

A more theoretical development for future work is functional calculation of the error bounds of the system. This model relies only on the proof that the resulting estimation $\hat{f}$ is bounded. Finding a relationship between the error tolerance, ϵ, and the estimation error, ζ, would strengthen the stability analysis as well as educate the designer on how to better pick gain values for the controller.

A possible endeavor in future of simulation is the use of an algorithm to provide a mapping of the C_p curve based on previous data points. The current error reduction method which uses the array throws out old values n+1 or greater. A new method could store the coordinates of λ, β, and C_p. Because we know that the nature of the C_p curve is a convex parabaloid-like function, these coordinates could be used to provide an estimate of the surface which could be maximized by some three dimensional gradient or least squares technique. The use of second order curve fitting could then be used to estimate the shape of the entire C_p surface. The two dimensional Lyapunov extremum seeking method could then be applied to the surface estimate.

Bibliography

[1] E. Hau, " Wind-turbines," *Springer*, Germany, 2000

[2] J. Layton, "How Wind Power Works," *science.howstuffworks.com*, [Online]. Available: science.howstuffworks.com/wind-power2.htm, [Accessed: June 17, 2010].

[3] V. Daniel Hunt, "Windpower: A Handbook on Wind Energy Conversion Systems," *Litton Educational Publishing Inc*, New York, 1981.

[4] "Parts of an electric generating windmill," *petervaldivia.com*, Oct. 24, 2008, [Online]. Available: http://www.petervaldivia.com/technology/energy/wind-energy.php, [Accessed: June 17, 2010].

[5] R. Swisher, C. R. De Auza, J. Clendenin, "Strong Winds on the Horizon: Wind Power Comes of Age," *Proceedings of the IEEE, Vol 89, No. 12*, December 2001.

[6] "Skystream 3.7 Photos," *skystreamcommercial.com*, [Online]. Available: http://skystreamcommercial.com/will-skystream-work/productphotos.php, [Accessed: June 17, 2010].

[7] A. Swapp, "Wind for Schools Project Power System Brief," May 2009. [Online]. Available: *http://www.windpoweringamerica.gov/schools_wfs_project.asp* [Accessed: June 14, 2010].

[8] "Turbine Overview," 2007. [Online]. Available: http://www.vestas.com/en/wind-power-plants/procurement/turbine-overview.aspx#/vestas-univers [Accessed: June 14, 2010].

[9] "2009 US Wind Industry Annual Market Report: Rankings," May 2010. [Online]. Available:*http://www.windpoweringamerica.gov/schools_wfs_project.asp* [Accessed: June 14, 2010].

[10] "Offshore Wind Energy," [Online]. Available: http://ocsenergy.anl.gov/guide/wind/index.cfm [Accessed: June 14, 2010]

[11] A. Bjartnes, "Dong, Scottish and Southern to Develop Dutch Wind Farms," *rechargenews.com*, Dec. 3, 2009, [Online]. Available: http://www.rechargenews.com/energy/wind/article200495.ece, [Accessed June 17, 2010].

[12] World Wind Energy Association, "World Wind Energy Report 2009," *9th World Wind Energy Conference and Exhibition on Large-Scale Integration of Wind Power*, Istanbul Turkey, June 15-17, 2010

[13] "World Wind Energy Report 2009," *wwindea.org*, March 10, 2009, [Online]. Available: http://www.wwindea.org/home/index.php?option=com_content&task=view&id=266&Itemid=2, [Accessed June 17, 2010].

[14] P. Costa, A. Martins, A. Carvalho, “Wind Energy Extraction and Conversion: Optimization through Variable Speed Generators and Non Linear Fuzzy Control,” *EWEC*, Athens Greece, March 2006.

[15] Kathryn E. Johnson, Lee J. Fingersh, Mark J. Balas, Lucy Y. Pao, “Methods for Increasing Region 2 Power Capture on a Variable-Speed Wind Turbine,” *Transactions of the ASME Vol. 126*, November, 2004.

[16] R. Datta and V. Ranganathan, “Variable speed wind power generation using a doubly fed wound rotor induction machine: A comparison with alternative schemes,” IEEE Trans. Energy Conversion, vol. 17, pp. 414–421, September, 2002.

[17] Anca D. Hansen, Florin Iov, Poul Sørensen, Frede Blaabjerg, “Overall Control Strategy of Variable Speed Doubly-Fed Induction Generator,” Nordic Wind Power Conference, Chalmers University, March, 2004.

[18] I. Munteanu, A. I. Bratcu, N Cutululis, E. Ceanga, “Optimal Control of Wind Energy Systems,” Springer-Verlag London Limited, 2008

[19] B. Malinga, J. Sneckenberger, J, Feliachi, “Modeling and Control of a Wind Turbine as a Distributed Resource,” Proc. 35th Southeastern Symp. Syst. Theory, Morgantown WV, 2003, pp. 108-112, 2003

[20] Brice Beltran, Tarek Ahmed-Ali, Mohamed El Hachemi Benbouzid, “Sliding Mode Power Control of Variable-Speed Wind Energy Conversion Systems,” IEEE Transactions on Energy Conversion Vol. 23, No. 2, June 2006

[21] Boubekeur Boukezzar, Mohamed M’Saad, “Robust Sliding Mode Control of a DFIG Variable Speed Wind Turbine for Power Production Optimization,” 16th Mediterranean Conference on Control and Automation, June 2008

[22] Kathryn E. Johnson, Lucy Y. Pao, Mark J Balas, Lee J. Fingersh, “Control of Variable-Speed Wind Turbines,” IEEE Control Systems Magazine, June 2006

[23] Xin Ma, “Adaptive Extremum Control and Wind Turbine Control,” Ph. D. Thesis Denmark Technical University, May 1997

[24] E. Iyasere, M. Salah, D. Dawson, J. Wagner, “Nonlinear Robust Control to Maximize Energy Capture in a Variable Speed Wind Turbine,” American Control Conference, June 2008

[25] E. Iyasere, M. Salah, D. Dawson, J. Wagner, “Nonlinear Robust Control to Maximize Energy Capture in a Variable Speed Wind Turbine Using a Separately Excited DC Generator,” College of Engineering and Science Control and Robotics Technical Report, June 2008

[26] Justin Creaby, Yaoyu Li, John E. Seem, “Maximizing Wind Turbine Energy Capture using Multivariable Extremum Seeking Control,” Wnd Engineering Vol. 33, No. 4, pp 361-388, 2009

[27] K. B. Ariyur, M. Kristic, “Real Time Optimization by Extremum Seeking Control,” John Wiley and Sons Inc., Hoboken, NJ, 2003

[28] Z. Chen, E. Spooner, “Grid Power Quality with Variable Speed Wind Turbines,” IEEE Transactions on Energy Conversion, Vol. 16, No. 2, June, 2001.

[29] Martin O. L. Hansen, “Aerodynamics of Wind Turbines,” EarthScan, London, UK, 2008, p. 4

[30] C. Makkar, G. Hu, W. G. Sawyer, W. E. Dixon, “Lyapunov-Based Tracking Control in the Presence of Uncertain Nonlinear Parameterizable Friction,” American Control Conference, June 2005

[31] B. Xian, D. M. Dawson, M.S. de Queiroz, J. Chen, “A Continuous Asymptotic Tracking Control Strategy for Uncertain Nonlinear Systems,” IEEE Transactions on Automatic Control Vol. 49, No. 7, July 2004

[32] Zillman, Donald, Walta, Mary E., Castiella, Inigo Del Guayo, “More Than Tilting at Windmills,” Washburn Law Journal Vol. 49, No. 1, pp 1-68, 2009

[33] F. Iov, A.D. Hansen, P. Sørensen, F. Blaabjerg, “Wind Turbine Blockset in Matlab/Simulink,” tech. rep. Aalborg University, Denmark, 2004.

[34] T. Hawkins, W. White, G. Hu, F. Darabi Sahneh, “Wind Turbine Power Capture Control with Robust Estimation,” ASME Dynamic Systems Control Conference, September 2010.

Appendix A - **Robust Estimator Controller Stability Proof**

The following development is the stability analysis for the robust estimator controller developed in Chapter 3. This analysis will prove that it provides semi-global asymptotic tracking of a desired trajectory. The proof of the controller is unique for this wind turbine application but is similar to that found in [30-31]. Recall the control laws given in equations (15) and (18) where

$$\tau_c = -J\dot{\omega}_d - J\alpha e - C_D\omega + \hat{f}(t) \tag{A1}$$

and

$$\begin{aligned}\hat{f}(t) &= (k_s+1)e(t) - (k_s+1)e(0) \\ &\quad + \int_0^t \left[(k_s+1)\alpha e(\tau) + \beta_c \mathrm{sgn}\big(e(\tau)\big)\right] d\tau.\end{aligned} \tag{A2}$$

Provided that β_c is selected according to the sufficient condition

$$\boldsymbol{\beta_c} > |-\dot{\boldsymbol{\tau}}_{\boldsymbol{aero}}| + \left|\frac{-\ddot{\boldsymbol{\tau}}_{\boldsymbol{aero}}}{\boldsymbol{\alpha}}\right| \tag{A3}$$

and term k_s is selected sufficiently large, a semi-global asymptotic result is produced. It must also be assumed that all closed loop system signals are bounded. This is a reasonable assumption because the source of energy to the wind capture system is the torque delivered to the drive shaft by the wind. To begin the stability analysis we must first develop an auxiliary term $P(t)$. P is defined as

$$P(t) = \beta_c|e(0)| - e(0)N_d(0) - \int_0^t L(\tau)d\tau. \tag{A4}$$

‘

The function L is also an auxiliary function which is defined as

$$L(t) = r(N_d(t) - \beta_c \mathrm{sgn}(e)). \tag{A5}$$

The time derivative of P is expressed as

$$\dot{P}(t) = -L(t) = -r(N_d(t) - \beta_c \mathrm{sgn}(e)). \tag{A6}$$

Provided that the sufficient condition of (A3) is met, the inequality

$$\int_0^t L(\tau)d\tau \leq \beta_c|e(0)| - e(0)N_d(0) \qquad \text{A(7)}$$

is obtained which proves that the term $P \geq 0$.

Let V be a positive definite Lyapunov candidate function defined as

$$V = \frac{1}{2}e^2 + \frac{1}{2}r^2 + P. \tag{A8}$$

Taking the time derivative of V gives

$$\dot{V} = e\dot{e} + r\dot{r} + \dot{P} \tag{A9}$$

or

$$\begin{aligned} \dot{V} = e(r - \alpha e) + r(-(k_s + 1)r - \beta_c \mathrm{sgn}(e) + N), \\ -r(N_d - \beta_c \mathrm{sgn}(e)) \qquad . \end{aligned} \tag{A10}$$

where substitutions were used for $\dot{e}$, $\dot{r}$, and $\dot{P}$ from (13), (20), and (A6), respectively. Expanding and simplifying expression in (A10), the Lyapunov time derivative equation becomes

$$\begin{aligned} \dot{V} = er - \alpha e^2 - (k_s + 1)r^2 - r\beta_c \mathrm{sgn}(e) \\ +rN - rN_d + r\beta_c \mathrm{sgn}(e) \qquad . \end{aligned} \tag{A11}$$

where

$$\dot{V} = -\alpha e^2 - (k_s + 1)r^2 + r\widetilde{N} \tag{A12}$$

and

$$\widetilde{N} = N - N_d. \tag{A13}$$

Using the mean value theorem, (A13) is can be written as

$$\widetilde{N} \leq \rho(\|z\|)\|z\| \tag{A14}$$

where ρ is a bounding function and z is given by

$$z = [e\ r]^T. \tag{A15}$$

Substituting (A14) into (A12) yields the relationship

$$\dot{V} \leq -\lambda\|z\|^2 - k_s r^2 + \rho(\|z\|)\|z\|r, \tag{A16}$$

where $\lambda = [\alpha\ 1]$. Note: λ in this analysis is not the same as the tip speed ratio. After completing the square, equation (A16) becomes

$$\dot{V} \leq -\lambda\|z\|^2 + \frac{\rho^2(\|z\|)\|z\|^2}{4k_s} \tag{A17}$$

or

$$\dot{V} \le -\left(\lambda - \frac{\rho^2(\|z\|)}{4k_s}\right)\|z\|^2.$$

The system is stable under the conditions that

$$k_s > \frac{1}{4\lambda}\rho^2\|z\| \quad \text{or} \quad \|z\| < \rho^{-1}(2\sqrt{\lambda k_s}). \tag{A18}$$

Appendix B - **Simulink Embedded Code and Subsystem Blocks**

Figure 5.1 shows the complete diagram of all of the subsystems used to simulate the wind turbine. This appendix will show in detail all of the information contained within those subsystem blocks and also detail the MATLAB code used in the blocks of each of the controllers.

Wind Turbine "Plant" Subsystems

This subsection will describe the Simulink code used to model the wind turbine. Also included in this section are the wind and noise generation subsystems. The Robust Controller and Lyapunov-based Extremum Seeking Controller will be detailed in the next section of this appendix. Let us begin with the wind generation block. Inside the subsystem the block diagram is shown in Figure B.1.

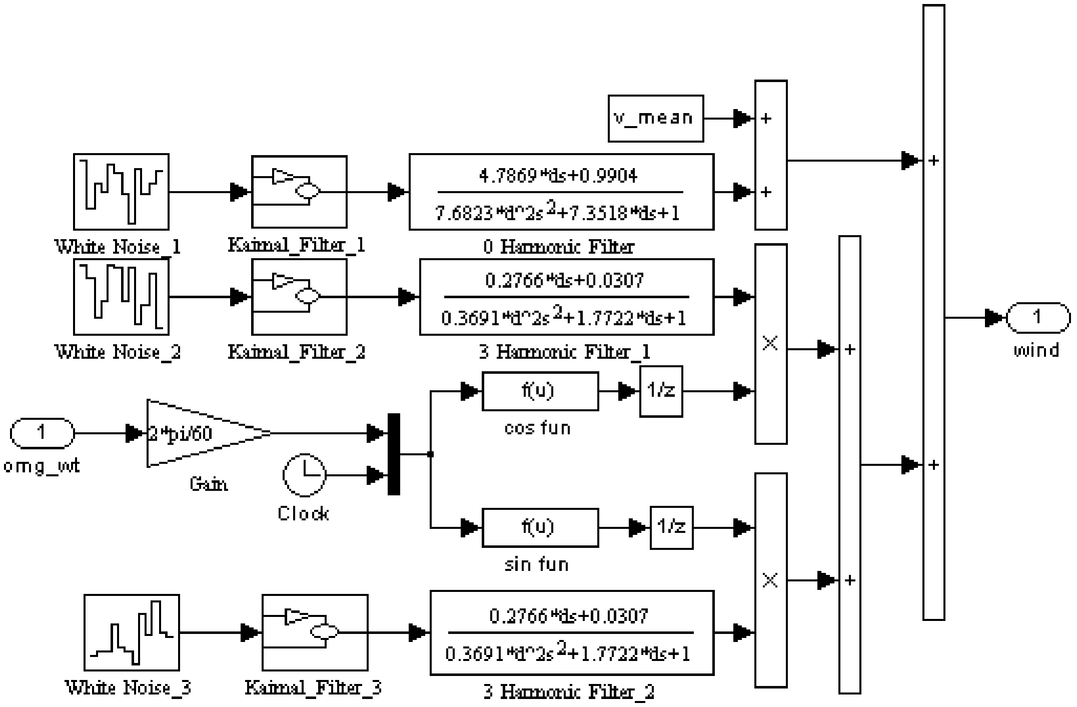

Figure B.1 Wind Generation Subsystem

This block uses white noise generators to generate random numbers and sends them through Kaimal filters which describe the statistical distribution of wind. This subsystem is part of the Wind Turbine Blockset [33].

Below, Figure B.2 shows the operation of the wind turbine rotor. This subsystem takes in the wind signal and the shaft angular velocity to show how to compute lambda. the blade pitch dynamics are also contained within this block. Because the blade pitch dynamics were neglected, the time constant for the transfer function representing these dynamics was set to zero. With the values of tip-speed ratio and blade pitch, this subsystem then uses a lookup table (defined with default data) to determine the value of C_q, linearly interpolating between data points when necessary. This value of C_q is then used to determine the aerodynamic torque of the wind on the rotor and this torque value is sent to the subsequent block of plant dynamics.

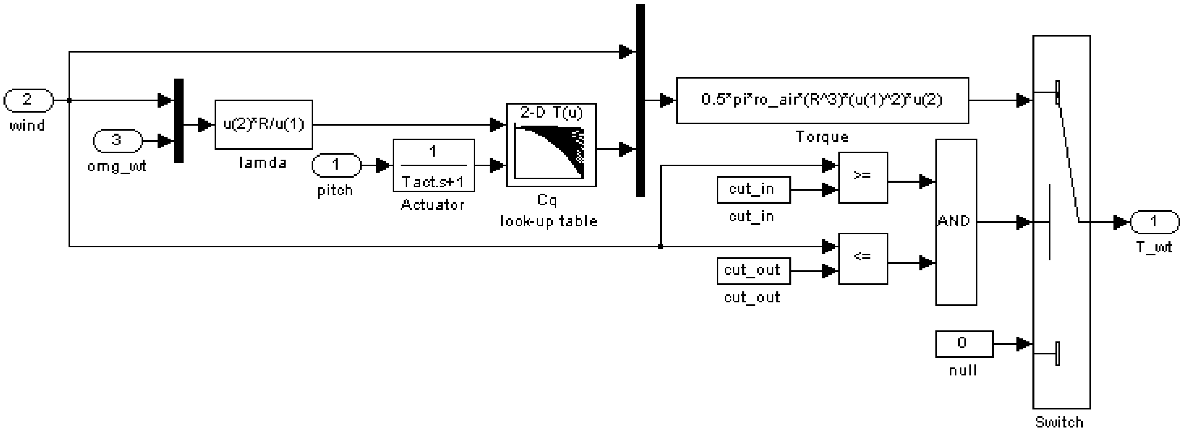

Figure B.2 Wind Rotor Subsystem

Figure B.3 below shows the Simulink code for the plant dynamics. The gain term K in the feedback loop represents the constant damping coefficient denoted in this paper by C_D. The gain term leading the integrator is the inertia, *J*, and the two inputs are the aerodynamic torque from the rotor and the control torque from the generator.

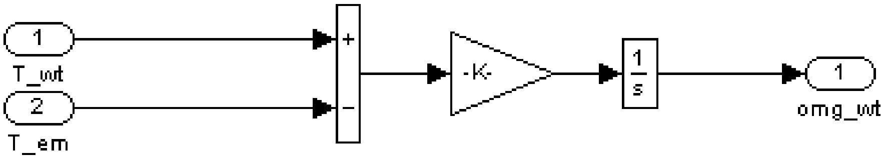

Figure B.3 Wind Turbine Plant Dynamics Subsystem

The wind velocity measurement noise was modeled by adding a zero mean Gaussian random number to the wind signal in the control feedback loop. This measurement noise was not delivered to the wind turbine dynamic system but used in the control calculations only. A figure will not be shown of the noise generation block because it is part of the standard Simulink source library.

Wind Turbine Controller Subsystems

This section includes the Simulink block diagrams and the associated MATLAB code for the controller used in the simulation. Two controller blocks were used: One for the robust controller and the other for the Lyapunov-based extremum seeking controller. Each of the subsystem blocks contains an embedded MATLAB function block which consists of custom written code that implements each of the control strategies. Following a description of the Simulink graphical code, a commented version of the code for each control block will be presented.

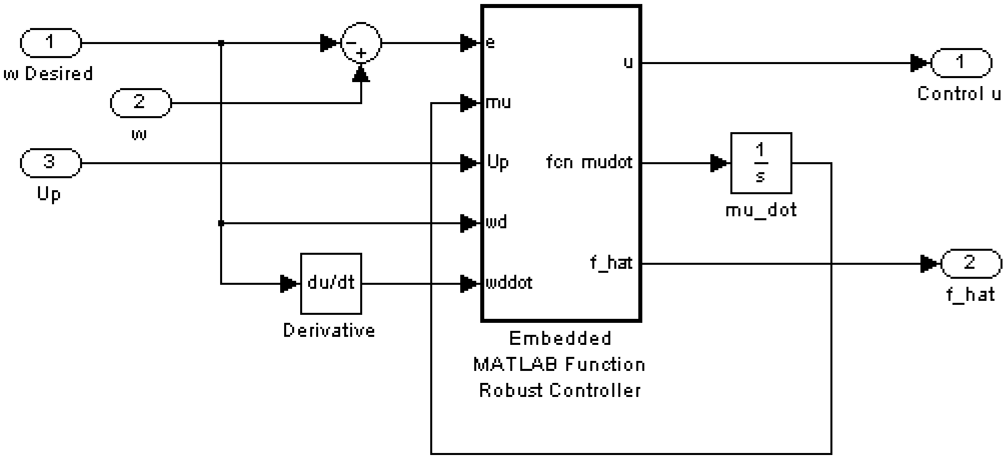

Figure B.4 Simulink Diagram for Robust Estimator Controller

Figure B.4 shows the Simulink diagram for the robust estimator controller. This takes in the values of shaft angular velocity and desired shaft angular velocity to compute a state tracking error signal. The input "Up" is the Boolean variable which gets set once a new set point is determined. This allows the e_0 term, initial error, in the robust controller to be re-initialized as the new trajectory point is set. The first output is the control torque, τ_c, denoted by u. The next is the output variable μ corresponding to the integral term of the robust controller. The expression within the integral is part of the output and this is integrated in time for the control law. Similarly, the parameter $\dot{\omega}_d$ is calculated by passing ω_d, decided by the Lyapunov controller, through a time derivative function determining $\dot{\omega}_d$ which is required for the computation of the robust control law, τ_c. The final output is $\hat{f}$ which is the estimate of the aerodynamic torque, $\underline{\tau_{aero}}$. The following MATLAB code is the custom function that is inside the Embedded MATLAB Function Robust Controller block.

Robust Estimator Controller Embedded MATLAB Code

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Robust Estimator Controller Block
%
%INPUTS: angluar velocity error, integrated fhat parameter, boolean update
%        desired angular velocity, time derivative of desired angular
%        velocity trajecotry.
%OUTPUTS: control torque, differential for fhat update, aerodynamic torque
%         estimation.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [u,mudot,f_hat] = fcn(e,mu,Up,wd,wddot)

% Definition of Constants
J=100000;                               % Rotor Inertia
Cd=1;                                   % Damping Coefficient

% Declaration of Persistent Variables
persistent e0 wdk1

% Definition of Controller Gains
k1=3000000; Bc=2; a=5;

% Initializations of Persistent Variables
if isempty(e0)
    e0=e;
    wdk1=wd;
end

% Error Update Between Discrete Steps
if Up==1
    e0=wd-wdk1;
    wdk1=wd;
end

% fhat Robust Law
f_hat=(k1+1)*(e-e0)+mu;
mudot=(k1+1)*a*e+Bc*tanh(e);

% Calculation of Omega (Angular Shaft Speed)
w=e+wd;

% Control Law
u=-J*wddot-J*a*e+f_hat;

% Generator Torque Saturation
umax=100e5; umin=0;

if u>umax
```

```
    u=umax;
end
if u<umin
    u=umin;
end

wd_k1=wd;

end
```

The block of Figure B.5 is the subsystem for the Lyapunov-based extremum seeking controller. This subsystem is responsible for several computations such as determining when the error has stabilized with tolerance, the estimation of C$_p$, the computation of the partial derivatives of $\hat{C}_p$ with respect to λ and β, and choosing the new set points of β and ω_d. The extremum seeking subsystem is shown below in Figure B.5.

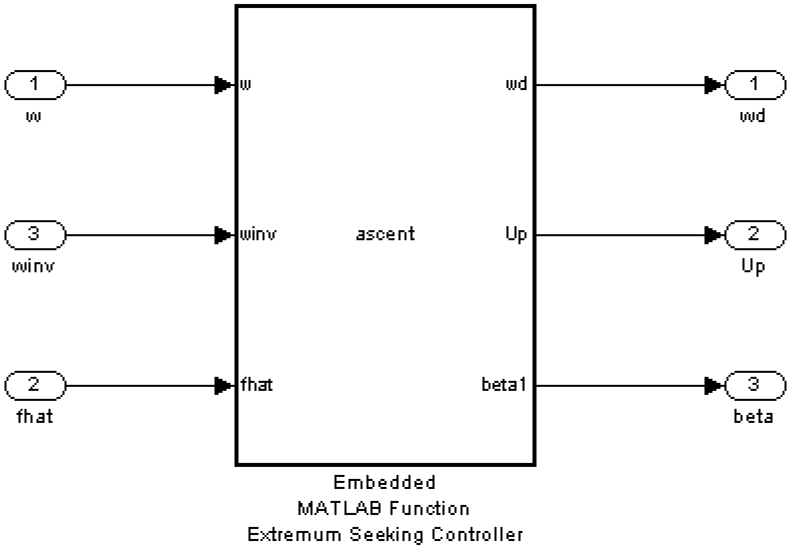

Figure B.5 Simulink Block Diagram for Lyapunov-Based Extremum Seeking Controller

The inputs of this block are the shaft angular velocity, the wind velocity (with measurement noise), and the estimate of τ_{aero}, $\hat{f}$. The outputs of this block are the Up Boolean variable. This parameter is set when a new value of λ_d is determined. The other outputs are the new desired value of ω_d and the new blade pitch, β. The code for this block is shown below.

Embedded MATLAB Code for the Lyapunov-Based Extremum Seeking Controller

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Trajectory Generation Function Block
%
%INPUTS: omega shaft, wind speed, estimated Taero (from PI), time
%OUTPUTS: omega desired, Update (Boolean Flag for PI), Cphat, dCp/dlambda
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [wd,Up,Cphat1,beta1,lamd1,Cphatavg1]  = ascent(w,winv,fhat,t)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Declaration of Persistent Variables
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

persistent beta betak1 count1 count2 Cphatarrlam Cphatarrbeta gamb Cphat
persistent dCpdbeta dCpdlam Cphatavg lamk2 lamholdup lamholddown flag gam
persistent Bool tk1 lamk1 count Cphatk1 lamd lamdk1 check



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Declaration of Constants
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

n=40;                       % Array Size
R=40;                       % Blade Radius (including hub)
A=pi*R^2;                   % Swept Area
p=1.25;                     % rho - Air Density
Up=0;                       % Update Boolean Flag Reset
lam=w*R/winv;               % Calculation of lambda from inputs omega and v
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Initialization of persistent variables for time t=0
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if isempty(count)
    gam=0.25;                       % Gamma sub lambda
    gamb=0.5;                       % Gamma sub beta
    lamholdup=0;                    % Error Tolerance Calc Variable
    lamholddown=0;                  % Error Tolerance Calc Variable
    count1=1;                       % Initialization of Counter
    count2=1;                       % Initialization of Counter
    count=1;                        % Initialization of Counter
    Cphatarrlam=zeros(n,2);         % Malloc Cp/lam Array
    Cphatarrbeta=zeros(n,2);        % Malloc Cp/beta Array
    dCpdbeta=0;                     % Init dCp/dbeta
    dCpdlam=0;                      % Init dCp/dlambda
    Cphatavg=0;                     % Init Cphat Average Calc to zero
    lamk2=0;                        % Init of lambda k-2 for Tolerance Test
    betak1=4;                       % Init of beta k-1 for finite diff
    beta=3;                         % Init of Inital beta Value
    lamk1=0;                        % Init of lambda k-1 for finite diff
    Cphatk1=0;                      % Init of Cphat k-1 for finite diff
    flag=1;                         % Lambda/Beta Selection Boolean Var
    lamd=1;%wdhold*R/winv;          % Init of Desired Lambda
    lamdk1=0.1;                     % Init of Desired Lambda k-1
    Bool=0;                         % Init of Boolean 'Within Tolerance'
    Cphat=.1;                       % Init of Cphat Value
    check=1;                        % Init of Rising/Falling Tolerance Bool
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Test for System Stability
%
% FUNCTIONALITY: The following code section computes whether the error of
%                system state omega is within tolerance epsilon. When this
%                it is within tolerance, fhat is suitable for estimation of
%                Cphat and Bool is set to 1.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Top Peak Check
if (lam > lamk2) && (check==1)
    lamholdup=lam;
else
    lamholdup=lamk2;
    check=2;
end

% Low Peak Check
if (lam < lamk2) && (check==2)
    lamholddown=lam;
else
    lamholddown=lamk2;
    check=1;
end

epsilon=0.001;                      % Tolerance boundary +-

if (abs(lamd-lamholdup) <= epsilon) && (abs(lamd-lamholddown) <= epsilon)
    Bool=1;
    lamholdup=0;
    lamholddown=0;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Extremum Seeking
%
% FUNCTIONALITY: Generates next desired set point of omega and beta.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if (Bool==1)                              % Check Stabilization Boolean

    Cphat=(fhat)*w/(0.5*p*A*winv^3);      % Calculate Estimation of Cp
    Cphatavg=Cphatavg+Cphat;              % Calculation of Avg Cphat

% Hyperbolic Tangent Gain Weighting Error Reduction
    A=0.75;                               % Vertical Shift
    B=0.75;                               % Vertical Stretch
    C=12;                                 % Horizontal Stretch
    D=4.5;                                % Horizontal Shift
    gam=A-B*tanh(C*Cphat-D);              % Gamma sub Lambda Gain Calc
    gamb=gam;                             % Gamma sub Bega Gain Calc

% Alternating Method Partial Derivative/Set-Point Computation
    if flag==1                            % Calc beta k+1 and dCp/dlam

        beta=betak1+gamb*dCpdbeta;        % Calculation of beta set-point

        if count1 < n                     % If array not full, finite diff
            dCpdlam=sign((Cphat-Cphatk1)/(lamd-lamk1));
            Cphatarrlam(count1,2)=Cphat;
            Cphatarrlam(count1,1)=lamd;
            dCpdt=0;
        elseif (count1 >= n) && (Cphat >= 0.45)
            Cphatarrlam(1:n-1,2)=Cphatarrlam(2:n,2);
            Cphatarrlam(n,2)=Cphat;
            Cphatarrlam(1:n-1,1)=Cphatarrlam(2:n,1);
            Cphatarrlam(n,1)=lamd;
            pol=polyfit(Cphatarrlam(:,1),Cphatarrlam(:,2),1);
            dCpdt1=polyder(pol);
            dCpdt=polyval(dCpdt1,lam);
            dCpdlam=dCpdt;
        else
            dCpdlam=sign((Cphat-Cphatk1)/(lamd-lamk1));
        end

        flag=0;                           % Toggle Boolean lam/beta Flag
        Up=1;                             % Send Update Boolean
        count1=count1+1;                  % Increment Lambda Array Counter

    else                                  % Calc lam k+1 and dCp/dbeta
        lamd=lamdk1+gam*dCpdlam;          % Calculation of Lambda Set Pt.
        if count2 < n                     % If Array not full, finit diff
            dCpdbeta=sign((Cphat-Cphatk1)/(beta-betak1));
            Cphatarrbeta(count2,2)=Cphat;
            Cphatarrbeta(count2,1)=beta;
            dCpdt=0;
        elseif (count >= n) && (Cphat >= 0.45)
            Cphatarrbeta(1:n-1,2)=Cphatarrbeta(2:n,2);
            Cphatarrbeta(n,2)=Cphat;
            Cphatarrbeta(1:n-1,1)=Cphatarrbeta(2:n,1);
            Cphatarrbeta(n,1)=beta;
            pol=polyfit(Cphatarrbeta(:,1),Cphatarrbeta(:,2),1);
            dCpdt1=polyder(pol);
            dCpdt=polyval(dCpdt1,beta);
            dCpdbeta=dCpdt;
        else
            dCpdbeta=sign((Cphat-Cphatk1)/(beta-betak1));
        end

        lamdk1=lamd;                      % Store k-1 Desired Lambda Value
        lamk1=lam;                        % Store k-1 Lambda Value
        flag=1;                           % Toggle Boolean lam/beta Flag
        betak1=beta;                      % Store k-1 Beta Value
        Up=1;                             % Raise New Traject Boolean
        count2=count2+1;                  % Increment Beta Array Counter
    end

    Cphatk1=Cphat;                        % Store k-1 Cphat Value
    Bool=0;                               % Reset Stability Tolerance Bool
    count=count+1;                        % Increment Average Counter

else                                      % If not stable, hold lamd set pt
    lamd=lamd;

end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Update t-1 variables and construct final output variables
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
wd=lamd*winv/R;                     % Calculating omega desired
lamdk1=lamd;                        % Store lambda desired k-1 step
beta1=beta;                         % Store beta k-1 step
lamd1=lamd;                         % Output Alias for Desired Lambda
lamk2=lam;                          % Store lambda k-2 step
Cphat1=Cphat;                       % Output Alias for Cphat
Cphatavg1=Cphatavg/count;           % Compute Average of Cphat

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end
```