Proceedings of the IMAC-XXVIII
February 1–4, 2010, Jacksonville, Florida USA

Force Estimation via Kalman Filtering for Wind Turbine Blade Control

Jonathan C. Berg[1] Dr. A. Keith Miller
Department of Mechanical Engineering
New Mexico Institute of Mining and Technology
801 Leroy Place, Socorro, NM 87801

ABSTRACT

Under simulation, the forces on a wind turbine blade are estimated in real time and then included in a control algorithm which seeks to manage loads and improve rotor performance. A Kalman filter, modified to include the input vector in the state vector, is employed to estimate the stochastic load state and full blade state from a small number of structural response measurements. Blade pitch and flap-like control surfaces are considered in a control scheme utilizing the estimated modal-equivalent forces. It is shown that the shape of the load profile can be regulated with such a controller, thereby reducing the undesirable effects of a wind gust.

1 BACKGROUND

In current technology, wind turbines are mostly passive structures. Control authority is limited to relatively slow moving actuators, and sensing is limited to what is required for essential control functions and electric grid integration. Research trends indicate that future generations will have active components which respond quickly to the wind environment, thereby improving efficiency and controlling loads [1].

Making the active components respond properly to the wind environment requires knowledge of either the local airflow conditions or the resulting aerodynamic forces on the blades. The first part of this paper proposes a method of estimating the forces which involves measuring the structural response and then using a modified Kalman filter to infer the stochastic wind load. The method allows a small number of sensors to produce a full state estimate. The second part of this paper discusses how the load estimate might be used to control blade pitch or an active aerodynamic device.

Input estimation of beam structures has been studied by C.K. Ma [2] using the method described by Tuan [3]. Their technique also uses the Kalman filter, but the excitation forces are estimated by a recursive least-squares algorithm that operates on information provided by the standard Kalman filter.

In regard to wind turbines, estimation of effective wind speed has been explored by X. Ma [4] and Østergaard [5]. Boukhezzar [6] uses the Kalman filter to estimate aerodynamic torque from the rotor speed and also demonstrates how to use the estimate for control purposes. Ehlers [7] looks at sensor selection and turbine state estimation for control purposes.

2 SYSTEM MODEL

The force estimation technique described in this paper begins with a linear structural model of the form

$$\mathbf{M}\ddot{\boldsymbol{\xi}} + \mathbf{C}\dot{\boldsymbol{\xi}} + \mathbf{K}\boldsymbol{\xi} = \mathbf{F}\mathbf{u} + \mathbf{F_d}\mathbf{w} \tag{1}$$

where $(^{\bullet})$ denotes the time derivative and $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{K}$ are respectively the mass, damping, and stiffness matrices. The variable $\boldsymbol{\xi}$ is the vector of displacement degrees-of-freedom and is also called the physical vector. The variable $\mathbf{u}$ is the vector

[1] Work in partial fulfillment of completion of a Masters of Science Degree in Mechanical Engineering, NMT.

of applied forces and can also be considered the vector of control inputs. Disturbance inputs to the system are represented by the vector $\mathbf{w}$. The matrices $\mathbf{F}$ and $\mathbf{F_d}$ are included to facilitate mapping of the inputs to the system. Second order differential equations result naturally from mechanical systems, however, the following first order description is more convenient for analysis.

The state space representation of a dynamic system has the form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{G}\mathbf{w} \tag{2}$$

where $\mathbf{x}$ is the vector of state variables, $\mathbf{u}$ is the control input vector, and $\mathbf{w}$ is the input disturbance (or process noise) vector. Matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{G}$ are respectively the state matrix, control input matrix, and input disturbance matrix. Physical measurements of the system are modeled as linear combinations of the state variables. The output equation is

$$\tilde{\mathbf{y}} = \mathbf{H}\mathbf{x} + \mathbf{v} \tag{3}$$

where $\tilde{\mathbf{y}}$ is the vector of measured outputs, $\mathbf{H}$ is the output matrix, and $\mathbf{v}$ is the sensor noise vector.

The state variable assignment for a given system is not unique. For the general mechanical system of equation (1), the most common choice is $\mathbf{x} = [\boldsymbol{\xi}^T \ \dot{\boldsymbol{\xi}}^T]^T$. The resulting system matrices are given in equation (4).

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C} \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1}\mathbf{F} \end{bmatrix} \qquad \mathbf{G} = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1}\mathbf{F_d} \end{bmatrix} \tag{4}$$

Equation (2) is continuous in time, meaning that the system progresses smoothly from one state to the next and the dynamics are described by differential equations. In certain situations such as simulation or control implementation, the system is sampled at regular time intervals, which makes a discrete-time description more convenient. The differential equations become recursive equations relating the state variables at integer multiples of the time step Δt. In the following notation, the subscript k refers to the kth time step, meaning $\mathbf{x}_k = \mathbf{x}(k\Delta t)$. Reference [8] shows that, when $\mathbf{A}$ and $\mathbf{B}$ are constant in time, the equations become

$$\mathbf{x}_{k+1} = \boldsymbol{\Phi}\mathbf{x}_k + \boldsymbol{\Gamma}\mathbf{u}_k + \boldsymbol{\Upsilon}\mathbf{w}_k \tag{5a}$$

$$\tilde{\mathbf{y}}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \tag{5b}$$

where

$$\boldsymbol{\Phi} = e^{\mathbf{A}\Delta t} \qquad \boldsymbol{\Gamma} = \left[\int_0^{\Delta t} e^{\mathbf{A}t}dt\right]\mathbf{B} \qquad \boldsymbol{\Upsilon} = \left[\int_0^{\Delta t} e^{\mathbf{A}t}dt\right]\mathbf{G}$$

In practice, the discrete-time system matrices can be found using a numerical approach. The MATLAB® command `c2d()`, which is part of the Control System Toolbox, implements one numerical solution and is used in this work.

```
[Phi,Gamma]=c2d(A,B,DT)
[Phi,Upsilon]=c2d(A,G,DT)
```

3 THE KALMAN FILTER

The Kalman filter is a recursive filter that can smooth measurements and estimate the complete state of a dynamic system. Its formulation is based in probability theory and stochastic system modeling, making it a good choice for application to the highly variable wind environment. If the stochastic properties of the measurement noise $\mathbf{v}$ and process noise $\mathbf{w}$ are well known, state estimates can be considered optimal in the sense of least-square estimation error.

Revisiting the discrete-time state space formulation given by equation set (5), the noise vectors are now described using statistical descriptions. The notation $\mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R}_k)$ indicates that $\mathbf{v}_k$ is a random vector with normal (Gaussian) distribution having zero mean and covariance matrix $\mathbf{R}_k$. It is assumed that $\mathbf{w}_k$ and $\mathbf{v}_k$ are both white-noise processes.

$$\mathbf{x}_{k+1} = \boldsymbol{\Phi}_k\mathbf{x}_k + \boldsymbol{\Gamma}_k\mathbf{u}_k + \boldsymbol{\Upsilon}_k\mathbf{w}_k, \quad \mathbf{w}_k \sim N(\mathbf{0}, \mathbf{Q}_k) \tag{6a}$$

$$\tilde{\mathbf{y}}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k, \quad \mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R}_k) \tag{6b}$$

3.1 Discrete-Time Linear Kalman Filter

Crassidis and Junkins [8] provide a thorough discussion of various forms of the Kalman filter. Equation set (7) is the discrete-time linear Kalman filter. The matrix $\mathbf{K}_k$ is known as the Kalman gain matrix, and it serves to improve the model's state prediction with information from the output measurement $\tilde{\mathbf{y}}_k$. Defining the estimation error as the difference between the state estimate $\hat{\mathbf{x}}_k$ and the true state $\mathbf{x}_k$, the matrix $\mathbf{P}_k$ is the covariance of the estimation error.

$$\hat{\mathbf{x}}_{k+1} = \mathbf{\Phi}_k \hat{\mathbf{x}}_k + \mathbf{\Gamma}_k \mathbf{u}_k + \mathbf{\Phi}_k \mathbf{K}_k \left[\tilde{\mathbf{y}}_k - \mathbf{H}_k \hat{\mathbf{x}}_k\right] \tag{7a}$$

$$\mathbf{K}_k = \mathbf{P}_k \mathbf{H}_k^T \left[\mathbf{H}_k \mathbf{P}_k \mathbf{H}_k^T + \mathbf{R}_k\right]^{-1} \tag{7b}$$

$$\mathbf{P}_{k+1} = \mathbf{\Phi}_k \mathbf{P}_k \mathbf{\Phi}_k^T - \mathbf{\Phi}_k \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k \mathbf{\Phi}_k^T + \mathbf{\Upsilon}_k \mathbf{Q}_k \mathbf{\Upsilon}_k^T \tag{7c}$$

3.2 Input Estimation

To estimate the input vector $\mathbf{u}$, one approach is to redesign the Kalman filter such that the input vector is included in the state vector. However, the dynamics relating $\dot{\mathbf{u}}$ to the state vector are likely unknown. This lack of knowledge can be modeled by setting $\dot{\mathbf{u}} = \mathbf{0}$ and then adding a random vector $\mathbf{z}$, thus making $\mathbf{u}$ constant except for model uncertainty. The discrete-time description becomes

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{u}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}_k & \mathbf{\Gamma}_k \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix} + \begin{bmatrix} \mathbf{\Upsilon}_k & \mathbf{0} \\ \mathbf{0} & \Delta t\, \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{w}_k \\ \mathbf{z}_k \end{bmatrix}, \quad \begin{bmatrix} \mathbf{w}_k \\ \mathbf{z}_k \end{bmatrix} \sim N\left(\mathbf{0}, \begin{bmatrix} \mathbf{Q}_{\mathbf{w}\,k} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{\mathbf{z}\,k} \end{bmatrix}\right) \tag{8a}$$

$$\tilde{\mathbf{y}}_k = \begin{bmatrix} \mathbf{H}_k & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix} + \mathbf{v}_k, \quad \mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R}_k) \tag{8b}$$

where Δt is the time step.

Thus, the extended Kalman filter for input estimation uses the following alterations to equation sets (6) and (7).

$$\begin{gathered} \mathbf{\Phi}_k^\star = \begin{bmatrix} \mathbf{\Phi}_k & \mathbf{\Gamma}_k \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \mathbf{\Gamma}_k^\star = \mathbf{0}, \quad \mathbf{\Upsilon}_k^\star = \begin{bmatrix} \mathbf{\Upsilon}_k & \mathbf{0} \\ \mathbf{0} & \Delta t\, \mathbf{I} \end{bmatrix}, \\ \mathbf{Q}_k^\star = \begin{bmatrix} \mathbf{Q}_{\mathbf{w}\,k} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{\mathbf{z}\,k} \end{bmatrix}, \quad \mathbf{H}_k^\star = \begin{bmatrix} \mathbf{H}_k & \mathbf{0} \end{bmatrix} \end{gathered} \tag{9}$$

4 SIMULATION SETUP AND RESULTS

4.1 Initial Testing

Before applying the technique to a full wind turbine simulation, a single blade was simulated by applying realistic forces to a non-rotating cantilever beam model. To produce realistic forces, the FAST wind turbine code [9] with AeroDyn subroutines [10] was employed to calculate blade forces on the Baseline 1.5 MW model provided with version 6.01 of FAST. Full-field turbulent wind input was generated by TurbSim [11] using the normal turbulence model, intensity level C, and Kaimal spectral model. A mean wind speed of 9 m/s was chosen, placing the turbine operating point in Region 2 (variable rotor speed with blade pitch held constant).

Aerodynamic forces calculated by AeroDyn were saved as text files and then loaded into MATLAB for testing of the estimation procedure. The force vector had 15 entries, corresponding to the 15 blade elements defined in the AeroDyn input file. Six separate wind files and responses were generated to provide a variety of load profiles and events.

The cantilever beam model in MATLAB was constructed from simple Euler-Bernoulli 2D beam elements. Mass and stiffness properties were taken from the 1.5 MW blade description and utilized along with modal damping to produce a tapered beam representative of the blade. The beam was likewise given 15 elements, and discrete-time simulation was carried out using the same time step as the pre-calculated force data.

The matrix $\mathbf{F}$ was defined to map element forces to nodal forces and moments. Process noise was not of great interest for this part of the testing, so the disturbance input mapping matrix $\mathbf{F}_\mathbf{d}$ was set equal to an identity matrix of appropriate size and

$\mathbf{w}_k$ was given small variance. Considering $\mathbf{Q}_{\mathbf{z}k}$, it was noted from equation (8a) that the statistical properties of $\mathbf{z}$ are related to those of the input vector. So, element forces from over 10 minutes of simulated turbine operation were taken to calculate the statistics of the input vector, specifically the covariance of $\mathbf{u}$. Testing the filter with $\mathbf{Q}_{\mathbf{z}k}$ equal to $\mathrm{Cov}(\mathbf{u})$ produced input estimates that were deemed acceptable.

Using this setup, a study of various measurement types, sensor locations, and sensor noise levels was conducted. Measurement type and location were defined in the output matrix $\mathbf{H}_k$, and noise level was selected with the variable R. In summary, the parameters of the Kalman filter were

$$\mathbf{Q}_{\mathbf{w}k} = 1.0 \cdot \mathbf{I} \tag{10a}$$

$$\mathbf{Q}_{\mathbf{z}k} = \mathrm{Cov}(\mathbf{u}) \tag{10b}$$

$$\mathbf{R}_{\mathbf{k}} = R \cdot \mathbf{I} \tag{10c}$$

Of all the measurement types that were studied, FAST can output bending moment most easily. The next section describes how the force estimation technique was applied with the full FAST model in the loop with a controller. Thus, a brief discussion of the bending moment results of the sensor study is included here.

At least two sensors were required to estimate both the shape and magnitude of the load profile. With $R = 10^{-4}$, estimation error was minimized when bending moment was measured at the blade root and at node 8 (53% span). Note that the noise level was relative to the output level, which was scaled so that the output magnitude was approximately equal to one. For bending moment, this scaling resulted in output units of 1×10^6 Newton-meters. Figure 1 provides a plot of force estimation error over time and a few snapshots of the true force profile (solid line) and estimated force profile (dotted line). The estimate is not perfect, but it does track the shape of the force profile.

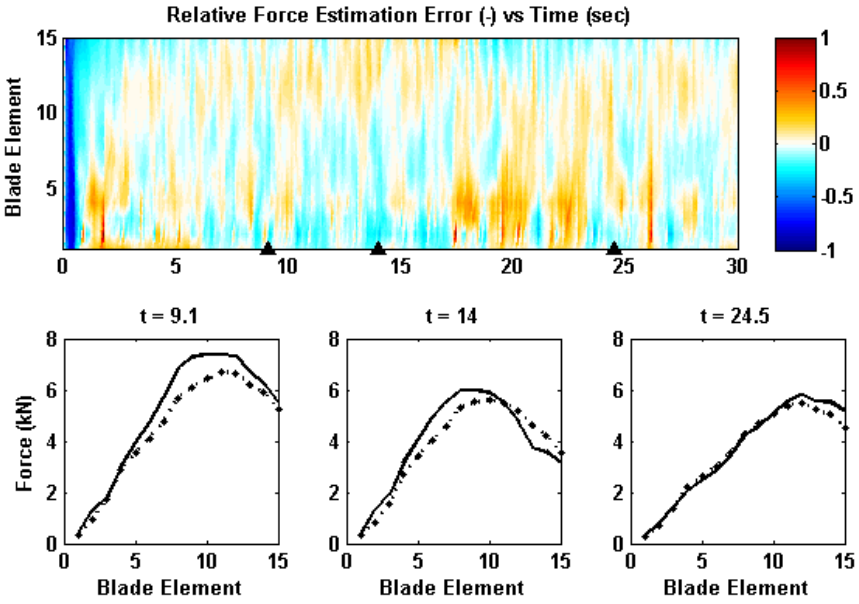

Figure 1: Top: force estimation error as a function of time and blade element. Bottom: the true force profile (solid) and estimated force profile (dotted) at three points in time.

4.2 Control Development

The technique was then applied inside MATLAB's Simulink environment with FAST included as an S-function block. With this configuration, the force estimate could be fed into a control algorithm while the simulation was running. The Kalman filter was implemented in MATLAB code as an S-function block.

It was found that a reduced-order model of the previous higher order beam model was useful for control purposes. First,

smaller system matrices resulted in a faster Kalman filter. Second, the 15 estimated forces were reduced down to a more manageable set of inputs for the control algorithm.

The form of the reduced model was given by

$$\mathbf{M_r}\ddot{\boldsymbol{\eta}} + \mathbf{C_r}\dot{\boldsymbol{\eta}} + \mathbf{K_r}\boldsymbol{\eta} = \mathbf{u_r} + \mathbf{w_r} \tag{11}$$

The transformation from physical coordinates to the reduced set of DOFs was accomplished by first finding the real-valued eigenvectors $\mathbf{V}$ of the mechanical system equations. A reduced set of bending modes $\mathbf{V_r}$ was then formed by selecting a few columns from $\mathbf{V}$. The eigenvectors had been scaled such that the reduced mass matrix equaled the identity matrix and the reduced stiffness matrix was diagonal and contained the squared natural frequencies.

$$\boldsymbol{\xi} = \mathbf{V_r}\boldsymbol{\eta} \tag{12a}$$
$$\mathbf{M_r} = \mathbf{V_r}^T\mathbf{M}\mathbf{V_r} = \mathbf{I} \tag{12b}$$
$$\mathbf{K_r} = \mathbf{V_r}^T\mathbf{K}\mathbf{V_r} = \lceil\omega^2\rfloor \tag{12c}$$
$$\mathbf{C_r} = \mathbf{V_r}^T\mathbf{C}\mathbf{V_r} = \lceil 2\zeta\omega\rfloor \tag{12d}$$
$$\mathbf{u_r} = \mathbf{V_r}^T\mathbf{F}\mathbf{u} \tag{12e}$$
$$\mathbf{w_r} = \mathbf{V_r}^T\mathbf{F_d}\mathbf{w} \tag{12f}$$

The state space form was then given by

$$\dot{\mathbf{x}}_\mathbf{r} = \mathbf{A_r}\mathbf{x_r} + \mathbf{B_r}\mathbf{u_r} + \mathbf{G_r}\mathbf{w_r} \tag{13a}$$
$$\tilde{\mathbf{y}} = \mathbf{H_r}\mathbf{x_r} + \mathbf{v} \tag{13b}$$

where $\mathbf{x_r} = [\boldsymbol{\eta}^T\ \dot{\boldsymbol{\eta}}^T]^T$ and

$$\mathbf{A_r} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{K_r} & -\mathbf{C_r} \end{bmatrix} \qquad \mathbf{B_r} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} \qquad \mathbf{G_r} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}$$
$$\mathbf{H_r} = \mathbf{H}\begin{bmatrix} \mathbf{V_r} & \mathbf{0} \\ \mathbf{0} & \mathbf{V_r} \end{bmatrix}$$

From this point, the procedure was the same as before. The continuous equations were converted to discrete time, and the Kalman filter's system matrices were augmented to estimate both the beam state and inputs as described previously.

To obtain the covariance of $\mathbf{w_r}$ in terms of $\mathbf{Q_w}$, the definition of covariance was applied to the random vector $\mathbf{w_r}$, resulting in

$$\mathbf{Q_{w_r}} = \mathbf{V_r}^T\mathbf{F_d}\mathbf{Q_w}\mathbf{F_d}^T\mathbf{V_r} \tag{14}$$

Likewise, the covariance of $\mathbf{u_r}$ was found to be

$$\mathrm{Cov}(\mathbf{u_r}) = \mathbf{V_r}^T\mathbf{F}\mathrm{Cov}(\mathbf{u})\mathbf{F}^T\mathbf{V_r} \tag{15}$$

In modal coordinates, the estimated input vector $\hat{\mathbf{u}}_\mathbf{r}$ can be compared directly to the true input vector $\mathbf{u_r}$. However, to compare results in physical coordinates, $\hat{\mathbf{u}}_\mathbf{r}$ must be related back to $\hat{\mathbf{u}}$. Although $\mathbf{V_r}^T$ is not square and can not be inverted exactly, the full set of eigenvectors is invertible. The resulting matrix $(\mathbf{V}^T)^{-1}$ is then reduced such that the remaining columns corresponded to the rows of $\hat{\mathbf{u}}_\mathbf{r}$.

$$\mathbf{F}\hat{\mathbf{u}} = [(\mathbf{V}^T)^{-1}]_\mathbf{r}\hat{\mathbf{u}}_\mathbf{r} \tag{16}$$

Thus, the estimated nodal forces $\mathbf{F}\hat{\mathbf{u}}$ computed by the above equation can be compared with the true nodal forces $\mathbf{F}\mathbf{u}$. Note that the columns of $(\mathbf{V}^T)^{-1}$ form a set of basis functions for the force profile. These basis functions have the unique property that when each is applied to the beam, the resulting static deformation is the corresponding mode shape. The first three force basis functions are plotted in Figure 2.

With the tools in place to estimate blade forces directly from the FAST outputs, a test case involving a wind gust was selected. In one particular wind input file, there was a point in the simulation where the forces at the blade tip dropped off drastically.

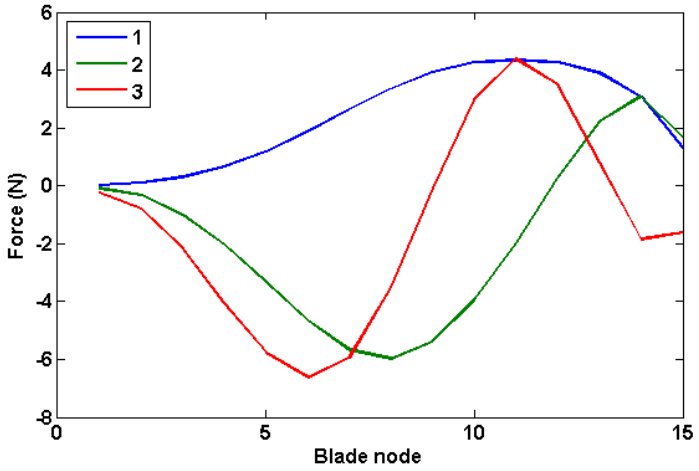

Figure 2: Force basis functions resulting from modal representation.

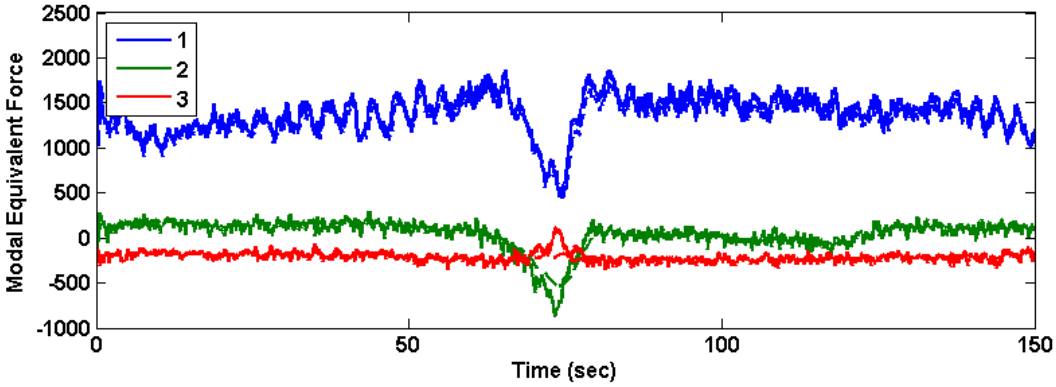

Figure 3: Modal equivalent forces showing a lull in the blade load. Dashed lines are the estimates.

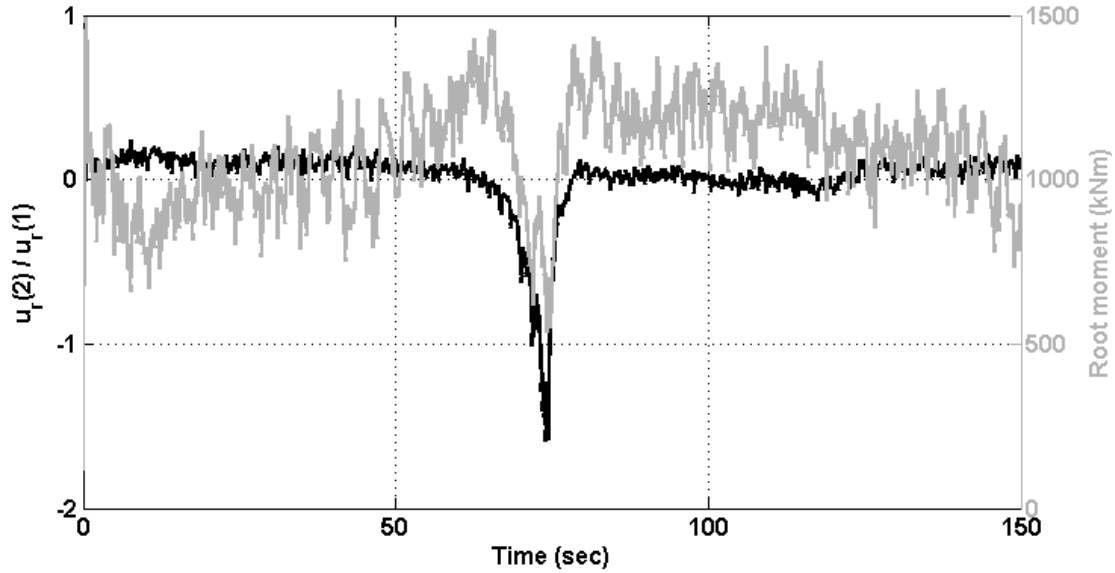

Figure 4: Ratio of the second modal force to the first. Root bending moment plotted for comparison.

A gust had caused the rotor to speed up and then the subsequent lull produced a drop in the tip load for a number of rotor revolutions. The lull can be seen in Figure 3 which plots the first three modal equivalent forces. The fact that the load profile had changed can be seen more clearly by plotting the ratio of the second modal force to the first (Figure 4).

If this ratio of the first and second modal forces were regulated by a controller, then the force profile would remain more constant. Having this control objective in mind, a simple proportional controller was defined which responded to this ratio whenever it fell below a threshold of 0.1. Equations (17) and (18) define the controller's input and output.

$$c_{in} = \hat{u}_{r,2}/\hat{u}_{r,1} - 0.1 \tag{17}$$

$$c_{out} = \begin{cases} K_P \cdot c_{in} & \text{for } c_{in} < 0 \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

The control scheme was evaluated using two forms of actuation: blade pitch and a morphing flap device modeled as in reference [12]. In both cases, all three turbine blades were controlled collectively with the same command signal rather than independently. For blade pitch, the output of the load controller was added to the output of the standard pitch controller (which was a constant 2.6 degrees). Tuning the load controller resulted in the gain $K_P = -12 \cdot \pi/180$ (pitch command in radians). For the morphing flap, the tuned gain was $K_P = 30$ (output in degrees of flap deflection). Figures 5 and 6 show the results for each control device. The maximum pitch action, above the 2.6 degree nominal value, was 1.6 degrees. The maximum flap deflection was 4.9 degrees. The controller's effect on generated power can be seen in Figure 7.

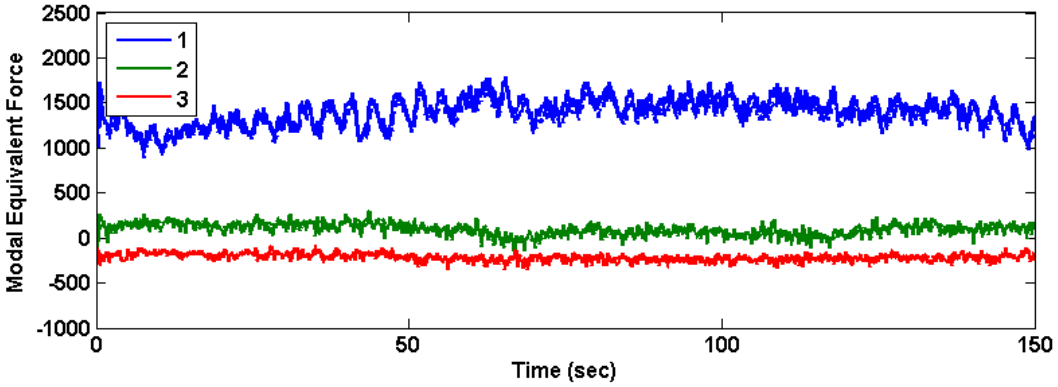

Figure 5: The lull in blade forces reduced by pitch action.

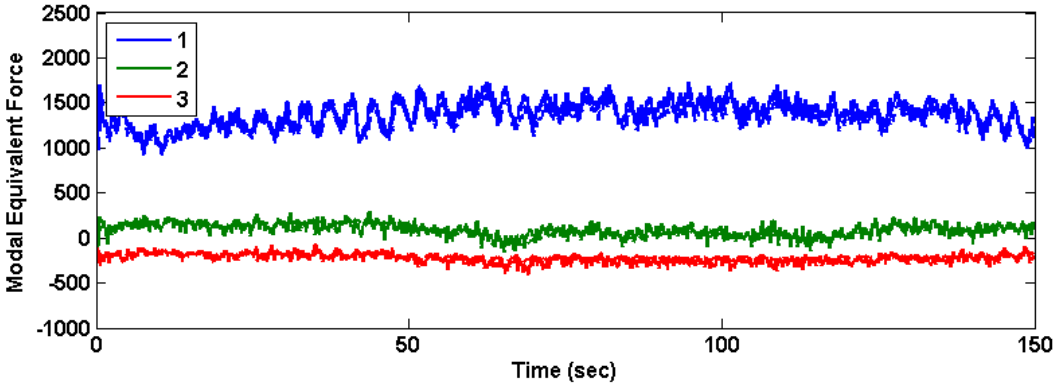

Figure 6: The lull in blade forces reduced by control of a morphing flap device.

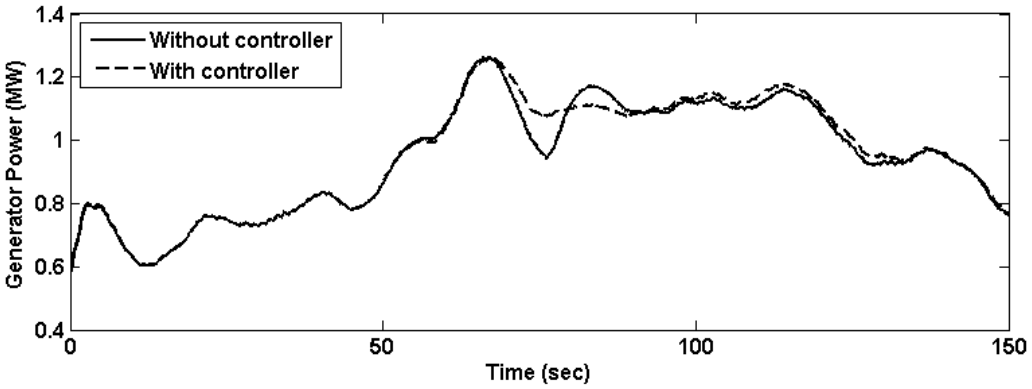

Figure 7: The controller's effect on generated power during the lull.

5 DISCUSSION

The purpose of this demonstration was to introduce the possibility of using blade force estimates in a control scheme. The approach taken here may not be the best solution when other factors are considered, and this controller has not been examined in terms of stability or robustness. However, the force estimation technique shows promise. Looking at Figure 4, the ratio of the modal forces provided more precise information than the root moment did by itself. Notice that the root moment increased before the lull, which would have set the controller going in the wrong direction if root moment were used to drive the controller. The ratio of modal forces remained rather constant until a control action was needed.

The peak-to-peak variation of root bending moment during the gust and lull was reduced by half, from 920 kN-m to 460 kN-m. Controlling large swings such as this is important for reducing fatigue. In addition, the power quality was improved by reducing the power fluctuation, and energy capture over the 150 second time frame was not hindered by the control action. Energy capture without the load controller was 39.38 kWh. With pitch and flap control, the numbers were 39.68 kWh and 39.78 kWh respectively.

In regard to pitch control versus active aerodynamic devices, the morphing flap device was used in this example in much the same way that the pitching action operates. The main difference was that pitch affected the entire blade span, while the morphing flap affected only the tip region where action was needed. A more advanced control scheme could target the effects of multiple active aerodynamic devices to specific parts of the blade.

Knowing the forces on each blade opens up the possibility of controlling directly for wind shear, yaw error, and pockets of turbulence hitting part of the rotor. These areas will be explored in future work.

ACKNOWLEDGMENTS

The authors would like to acknowledge the Wind & Water Power Technologies department of Sandia National Laboratories for providing access to the wind turbine model with morphing-flap modifications and for supporting this work.

Sandia National Laboratories is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

REFERENCES

[1] Johnson, S.J., van Dam, C.P., and Berg, D.E. Active load control techniques for wind turbines. Technical Report SAND2008-4809, Sandia National Laboratories, August 2008.

[2] Ma, C.K., Chang, J.M., and Lin, D.C. Input forces estimation of beam structures by an inverse method. *Journal of Sound and Vibration*, 259(2):387–407, January 2003.

[3] Tuan, P., Ji, C., Fong, L., and Huang, W. An input estimation approach to on-line two-dimensional inverse heat conduction problems. *Numerical Heat Transfer, Part B: Fundamentals*, 29(3):345–363, April 1996.

[4] Ma, X., Poulsen, N.K., and Bindner, H. Estimation of wind speed in connection to a wind turbine. Technical Report IMM-1995-26, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, December 1995.

[5] Østergaard, K.Z., Brath, P., and Stoustrup, J. Estimation of effective wind speed. *Journal of Physics: Conference Series*, 75, 2007.

[6] Boukhezzar, B., Siguerdidjane, H., and Hand, M. Nonlinear control of variable speed wind turbines for load reduction and power optimization. In *44th AIAA Aerospace Sciences Meeting and Exhibit*, pages 602–615, Reno, NV, January 2006.

[7] Ehlers, J., Diop, A., and Bindner, H. Sensor selection and state estimation for wind turbine controls. In *45th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, January 2007.

[8] Crassidis, J.L. and Junkins, J.L. *Optimal Estimation of Dynamic Systems*. Chapman & Hall/CRC, 2004.

[9] NWTC Design Codes. FAST by Jason Jonkman. http://wind.nrel.gov/designcodes/simulators/fast/, August 12, 2005. Accessed 21-May-2008.

[10] NWTC Design Codes. AeroDyn by Dr. David J. Laino. http://wind.nrel.gov/designcodes/simulators/aerodyn/, June 28, 2005.

[11] NWTC Design Codes. TurbSim by Neil Kelley, Bonnie Jonkman. http://wind.nrel.gov/designcodes/preprocessors/turbsim/, April 4, 2008. Accessed 28-May-2008.

[12] Wilson, D.G., Berg, D.E., Barone, M.F., Berg, J.C., Resor, B.R., and Lobitz, D.W. Active aerodynamic blade control design for load reduction on large wind turbines. In *European Wind Energy Conference & Exhibition*, March 2009.