



Hur, S. and Leithead, W. E. (2016) Model predictive and linear quadratic Gaussian control of a wind turbine. Optimal Control Applications and Methods. ISSN 0143-2087 , <http://dx.doi.org/10.1002/oca.2244>

This version is available at <http://strathprints.strath.ac.uk/55448/>

Strathprints is designed to allow users to access the research output of the University of Strathclyde. Unless otherwise explicitly stated on the manuscript, Copyright © and Moral Rights for the papers on this site are retained by the individual authors and/or other copyright owners. Please check the manuscript for details of any other licences that may have been applied. You may not engage in further distribution of the material for any profitmaking activities or any commercial gain. You may freely distribute both the url (<http://strathprints.strath.ac.uk/>) and the content of this paper for research or private study, educational, or not-for-profit purposes without prior permission or charge.

Any correspondence concerning this service should be sent to the Strathprints administrator: strathprints@strath.ac.uk

Model Predictive and Linear Quadratic Gaussian Control of a Wind Turbine

S. Hur* and W. E. Leithead

Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, G1 1XZ, UK

SUMMARY

Model Predictive and Linear Quadratic Gaussian controllers are designed for a 5MW variable-speed pitch-regulated wind turbine for three operating points – below rated wind speed, just above rated wind speed, and above rated wind speed. The controllers are designed based on two different linear dynamic models (at each operating point) of the same wind turbine to study the effect of utilising different control design models (i.e. the model used for designing a model-based controller) on the control performance. The performance of the LQG controller is enhanced by improving the robustness, achieved by replacing the Kalman filter with a modified Luenberger observer, whose gain is obtained to minimise the effect of uncertainty and disturbance. Copyright © 2015 John Wiley & Sons, Ltd.

Received ...

KEY WORDS: Wind turbine control, Model Predictive Control, Linear Quadratic Gaussian, observer

1. INTRODUCTION

There is much interest in renewable energy due to concern over the environment, and wind is considered to be one the most promising renewable energy sources, partly because wind is an infinite and free source of energy with no harmful waste products. A wind turbine converts the kinetic energy from the wind into mechanical energy. It is then converted into electricity, which is subsequently transmitted to a power grid. The turbine components responsible for these energy conversions are the rotor, consisting of the turbine hub and blades, and the nacelle mounted on top of the tower as shown in Figure 1. Inside the nacelle of a variable speed wind turbine [1, 2] are low-speed shaft, gearbox, high-speed shaft, and the generator. The low-speed shaft is coupled with the hub and the high-speed shaft drives the generator; that is, the wind blows onto the blades, the hub rotates due to aerodynamic forces, and this rotation is then sent through the gearbox to increase the generator speed. This paper is concerned with a 5MW variable-speed pitch-regulated horizontal-axis wind turbine, having three blades. The yaw mechanism, which is responsible for orientating the turbine towards the wind, is ignored in this paper.

The design of a control system for regulating variable-speed wind turbines involves both the determination of the operating strategy of the controller and its synthesis [3, 4]. The synthesis of the controller design is concerned with designing single input single output (SISO) linear controllers at different operating points. More specifically, linear controllers are often designed at 4 operating points, in the lowest wind speeds (mode 1), in intermediate wind speeds (mode 2), in higher, but still below rated, wind speeds (mode 3) and in above rated wind speeds (mode 4). Note that the rated wind speed is a pre-determined wind speed, at which the limit on power output to the wind turbine

*Correspondence to: Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, UK.
Email: hur.s.h@ieee.org

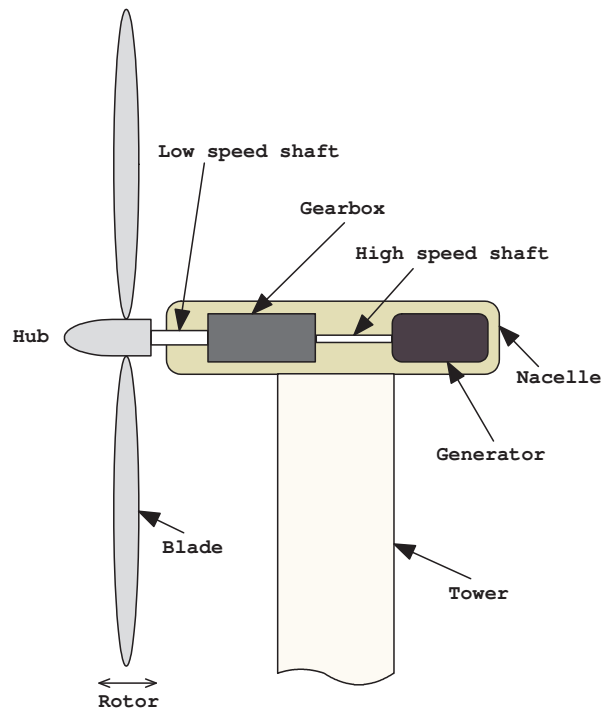


Figure 1. Main elements of a horizontal-axis variable-speed wind turbine.

(5MW in this paper), is reached. When the wind speed exceeds rated, the excess power in the wind is discarded to prevent the turbine from overloading.

Operating in mode 4 and switching between modes 3 and 4 are more challenging, and thus this paper focuses on modes 3 and 4. Since the operation is weakly nonlinear in mode 3, designing a linear controller at any mean wind speed within mode 3 suffices. In mode 4, as the operation is more strongly nonlinear, more than one linear controller is designed, and the one that performs best is chosen when designing the full nonlinear controller at a later stage. In mode 3, the linear controller is designed at a mean wind speed of 10 m/s. In modes 4, the linear controllers are designed at mean wind speeds of 12 and 14 m/s here. In each mode, a linear controller can be designed using any appropriate control method. The most common one stems from PI control [5, 6] (usually with significant modifications to incorporate fatigue reduction, anti-windup, etc.). In this paper, Model Predictive Control (MPC) [7, 8] and Linear Quadratic Gaussian (LQG) [9, 10] control are tested. Note that other control methods such as H_∞ [11] could also be equally appropriate. To design such model-based linear controllers in each mode, a control design model [12] is required in each mode; that is, based on the control design model, a linear controller is designed and its performance tested by application to the control design model before being applied to the turbine by being de-tuned.

The strategy of the controller design is responsible for switching between the linear controllers. To determine the strategy, the implementation issues such as accommodation of the variation in plant dynamics over the operational envelope, switching transients and actuator constraints, which are most relevant to the application, need to be identified and the controller realisation, which best resolves them, selected [3]. It is related to nonlinear aspects of the plant dynamics and an investigation of the global behaviour of the system is required. The linear controllers, as a result of the synthesis of the controller design, are combined by an appropriate switching method, yielding a global nonlinear controller that covers the full operational envelop, i.e. full envelop controller.

It is feasible to utilise MPC/LQG for both the synthesis and strategy, but strong nonlinearity that the wind turbines exhibit poses a significant challenge, resulting in a number of issues that need to be addressed. The controllers at each operating point have not only different gains, but also

different structures; that is, the controller in mode 2 exploits generator torque to regulate the power output while the controller in mode 3 pitches the blades to limit the power output. It thus requires complete switch-over of the controllers. The use of both generator torque and pitch in each mode is not feasible due to the physical constraints. Also, due to actuator constraints, switching has to be bumpless, and other issues such as fatigue reduction, anti-windup, etc., needs to be dealt with. Some research has been conducted [13, 14, 15] addressing these challenges, but this topic is beyond the scope of this paper, and MPC/LQG is only exploited for the synthesis here similarly to [16, 17, 18]; that is, linear MPC/LQG controllers are designed in each mode and subsequently gain-scheduled.

The overall control design steps in operation and pursued in this study can be briefly summarised as (Steps 1, 2 and 3 have thus far been described):

1. Development of linear control design models at each mode
2. Design of control synthesis based on the design models from Step 1 at each mode
3. Design of control strategy (i.e., gain-scheduling of the controllers from Step 2, incorporation of a drive-train damper, etc.) to obtain a full-envelope controller
4. Detuning of the full-envelope controller (from Step 3) by application to the high-fidelity aero-servo-elastic model, developed DNV-GL Bladed (Bladed), from which the linear models have been derived
5. Detuning and application of the full-envelope controller (from Step 4) to the real-life turbine that the aero-elastic model represents

This paper focuses only on Step 2 – and Step 1 as a prerequisite– and the following steps are beyond the scope of this paper. For Step 2, two different control design models are tested for designing MPC and LQG controllers to investigate the effect of exploiting different control design models on the control performance. Both control design models are linear models of the Supergen Wind Energy Technologies Consortium (Supergen) 5MW exemplar turbine.

The first linear model is obtained from the (nonlinear) Bladed model of the turbine using its built-in linearisation tool. Linear models are, in fact, identified as it generates input and state perturbations, and records the resulting variations in the state derivatives and selected outputs to finally derive a linearised model of the turbine in state-space form [19].

The second model is obtained by linearising the nonlinear model provided in [5, 20], with the parameters of the same turbine, by the use of the standard linearisation technique via symbolic differentiation (Taylor series expansion). Both linearised models are implemented in Matlab/SIMULINK® to allow the controller design to be performed in Matlab/SIMULINK. Throughout the paper, the Bladed linearised models are referred to as “Models A”, and the linearised models via symbolic differentiation “Models B”. *These linearised models could be provided by the authors upon request; please contact the corresponding author.*

When model-based controllers are designed based on a model (Step 1), the controllers are first tuned by application to the controller design model (Step 2). For the same reason here, when Models A are exploited as the controller design models, the controllers are tuned by application to Models A, and when Models B are exploited as the controller design models, the controllers are tuned by application to Models B.

Thus the controllers designed based on Models A (i.e. Models A-based controllers) and the controllers designed based on Models B (i.e. Models B-based controllers) are independent, and there is no direct connection between them. However, merely for the purpose of comparing the performance of these two controllers, the controllers need to be applied to another linearised model that is neither Model A nor Model B. Unfortunately, such a model is not available, and at this stage of the controller design the nonlinear turbine model, from which Models A and B have been derived, cannot be utilised as the strategy part (Step 3) of the controller design has not been dealt with. If the linear controllers are applied to the nonlinear model without completing Step 3, when the nonlinear model requires the linear controller to switch around at 12 m/s (the rated wind speed), the process would become unstable. For example at 12 and 14 m/s, the nonlinear model would require the linear controller to provide negative control action, i.e., pitch angle, that is not physically feasible, leading to actuator saturation. Also, drive-train and tower dampers [5] need to be designed as part of Step 3, or the process could become unstable.

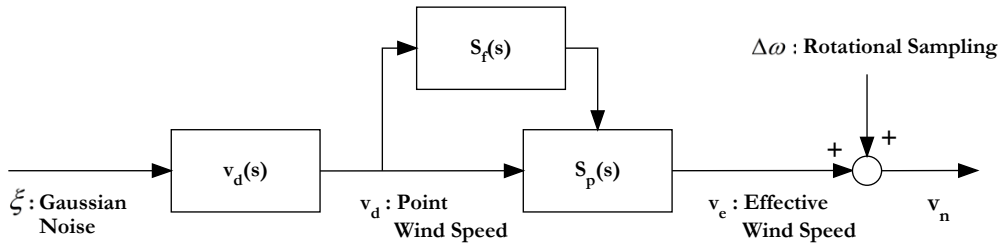


Figure 2. Point wind speed, effective wind speed, and rotational sampling.

Instead, the controllers are compared by both being applied to Models A, which have been chosen over Models B for being higher order, thereby possessing closer complexity to the real-life turbine. Since only Models B-based controllers would unfairly experience process-model mismatch, Models A-based controllers are expected to produce superior results, but, for designing MPC controllers, this is not true as demonstrated in Section 3; that is, the Models B-based controllers outperform Models A-based controllers despite the disadvantage.

It is also important to point out that, for designing MPC controllers, when both the controllers are applied to Models B, the Models B-based controllers outperform Models A-based controllers. This outcome is less significant as the Models B-based controllers have an edge over the Models A-based controllers since the Models A-based controllers are the only controllers experiencing process-model mismatch. Hence, this test in which both the controllers are applied to Models B is not presented in this paper, and only the test in which both the controllers are applied to Models A is reported.

For LQG controllers, different conclusions are drawn in Section 4.

The first contribution of this paper is to investigate the effect of the control design model on the performance of the controllers, i.e. the effect of utilising Models A in comparison to Models B when designing MPC and LQG controllers. The second contribution of this paper is to improve the performance of the LQG controller by replacing the Kalman filter with a modified version of the Luenberger observer [21, 22], whose gain is obtained to reduce the effect of uncertainty and noise.

The wind speed model, which is required for simulations, Models A, and Models B are reported in Section 2. Sections 3 and 4, respectively, report on the MPC and LQG controllers including some simulation results. In Section 5, the performance of the LQG controller is improved by replacing the Kalman filter with a modified version of the Luenberger observer. Conclusions are drawn in Section 6.

2. WIND SPEED AND LINEAR MODELS

2.1. Wind Speed Model

The wind is stochastically varying with time and continuously interacting with the rotor. The effective wind speed is wind speed averaged over the rotor area such that the spectrum of aerodynamic torque remains unchanged. It can be obtained by filtering the point wind speed. The power spectrum for the point wind speed is the Von Karman spectrum [23]

$$S_v(\omega) = 0.476\sigma_v^2 \frac{\frac{L_t}{\bar{V}}}{(1 + (\frac{\omega L_t}{\bar{V}})^2)^{5/6}} \quad (1)$$

where $L_t = 6.5h$ denotes the turbulence length of the spectrum, h height, and \bar{V} mean wind speed. σ_v represents turbulence intensity and is assumed to be 18.34, 17.03, and 16.10 % for mean wind speeds of 10, 12, and 14 m/s, respectively.

The Von Karman spectrum can be approximated by the following Dryden spectrum:

$$S_D(\omega) = \frac{1}{2\pi} \frac{b_d^2}{\omega^2 + \alpha_d^2} \quad (2)$$

The corresponding point wind speed is modelled by coloured noise as follows:

$$v_d = \frac{\bar{V}b_d}{s + a_d} \xi \quad (3)$$

where ξ denotes Gaussian noise.

For (3), the values of a_d and b_d , for which the Dryden spectrum best approximates the Von Karman spectrum, are

$$a_d = 1.14 \frac{\bar{V}}{L_t} \quad (4)$$

$$b_d = \sigma_v \sqrt{2a_d} \quad (5)$$

The effective wind speed is the wind speed actually experienced by the wind turbine. It can be modelled by spatially filtering point wind speed using the following filter:

$$S_p(s) = \frac{\sqrt{2}(\sqrt{2} + \sigma(\hat{V})s)}{(\sqrt{2} + \sqrt{\alpha}\sigma(\hat{V})s)(\sqrt{2} + \sigma(\hat{V})s/\sqrt{\alpha})} \quad (6)$$

α is set to 0.55 [24], and \hat{V} denotes the average wind speed over a period of time equivalent to a small number of rotations of the rotor. It can be derived by filtering the point wind speed through the first order filter

$$S_f(s) = \frac{1}{\tau s + 1} \quad (7)$$

τ is the period over which the wind speed is averaged, and $\sigma(\hat{V})$ is given as follows:

$$\sigma = \frac{\gamma R}{\hat{V}} \quad (8)$$

where γ denotes the turbulent wind field decay factor and is set to 1.3 [24]. The result from filtering the point wind speed through the spatial filter is shown for a mean wind speed of 10 m/s in Figure 3. The figure depicts the effective wind speed (in blue) together with the point wind speed (in red). Similar results can be obtained for mean wind speeds of 12 and 14 m/s. The rotor radius (R) and the height of the Supergen 5MW turbine are 60 m and 90 m, respectively.

Nonlinear rotational sampling [25] is subsequently added to the effective wind speed as illustrated in Figure 2. The equations for the rotational sampling ($\Delta\omega$) are summarised as follows:

$$\Delta\omega = \frac{1.25}{s/(3\Omega_o) + 1.25} x_r \quad (9)$$

where Ω_o denotes the operating rotor speed (i.e., 1.23 rad/s), x_r is given as

$$x_r = \epsilon_1 \cos(3\Omega_o t) + \epsilon_2 \cos(3\Omega_o t) \quad (10)$$

and

$$\dot{\epsilon}_1 = -a\epsilon_1 + b\xi_1 \quad (11)$$

$$\dot{\epsilon}_2 = -a\epsilon_2 + b\xi_2 \quad (12)$$

ξ_1 and ξ_2 denote Gaussian white noise, and a and b are, respectively, set to 0.4 and 3.

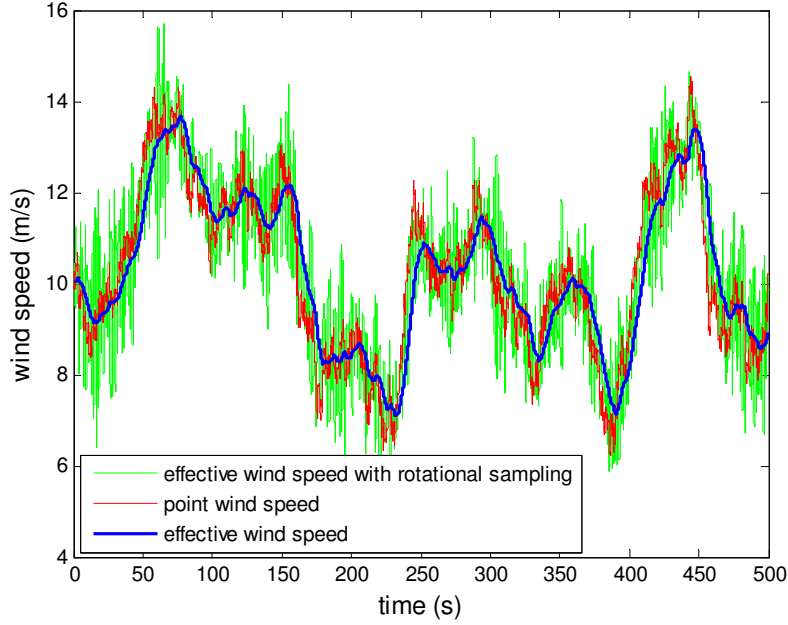


Figure 3. Point wind speed (red) vs effective wind speed (blue) vs effective wind speed with linear rotational sampling (green) at a mean wind speed of 10 m/s.

Throughout the paper, linearised turbine models are exploited to simulate the turbine. Hence, to be consistent, the wind speed model (including rotational sampling) is also linearised. The linear transfer functions for mean wind speeds of 10, 12, and 14 m/s, i.e. from ξ to v_n in Figure 2, are, respectively, as follow:

$$F_{10}(s) = \frac{3.962s^3 + 3.673s^2 + 1.564s + 0.187}{s^5 + 0.899s^4 + 15.930s^3 + 6.273s^2 + 0.632s + 0.010} \quad (13)$$

$$F_{12}(s) = \frac{3.963s^3 + 4.423s^2 + 2.281s + 0.328}{s^5 + 0.978s^4 + 15.990s^3 + 7.533s^2 + 0.910s + 0.017} \quad (14)$$

$$F_{14}(s) = \frac{3.964s^3 + 5.183s^2 + 3.154s + 0.532}{s^5 + 1.058s^4 + 16.050s^3 + 8.793s^2 + 1.239s + 0.028} \quad (15)$$

The wind speeds as a result of exploiting these transfer functions (i.e. effective wind speed with rotational sampling) are depicted in comparison to the point and effective wind speed in Figure 3.

2.2. Wind Turbine Linear Models A (Bladed Linearisation)

Dynamic models are linearised from the (nonlinear) Bladed model of Supergen 5MW exemplar turbine for three different operating points, below rated wind speed (10 m/s), just above rated wind speed (12 m/s), and above rated wind speed (14 m/s). In state space form, they have the following form:

$$\begin{aligned} \Delta \dot{\mathbf{x}}(t) &= A\Delta \mathbf{x}(t) + B\Delta \mathbf{u}_T(t) \\ \Delta \mathbf{y}(t) &= C\Delta \mathbf{x}(t) + D\Delta \mathbf{u}_T(t) \end{aligned} \quad (16)$$

where A , B , C , and D denote the state space matrices. $\Delta \mathbf{y}(t) \in \mathbb{R}^n$, $\Delta \mathbf{u}_T(t) \in \mathbb{R}^m$ and $\Delta \mathbf{x}(t) \in \mathbb{R}^r$ (where n and m are respectively 10 and 3 at each mean wind speed, and r is 30 at 10 m/s and 26 at

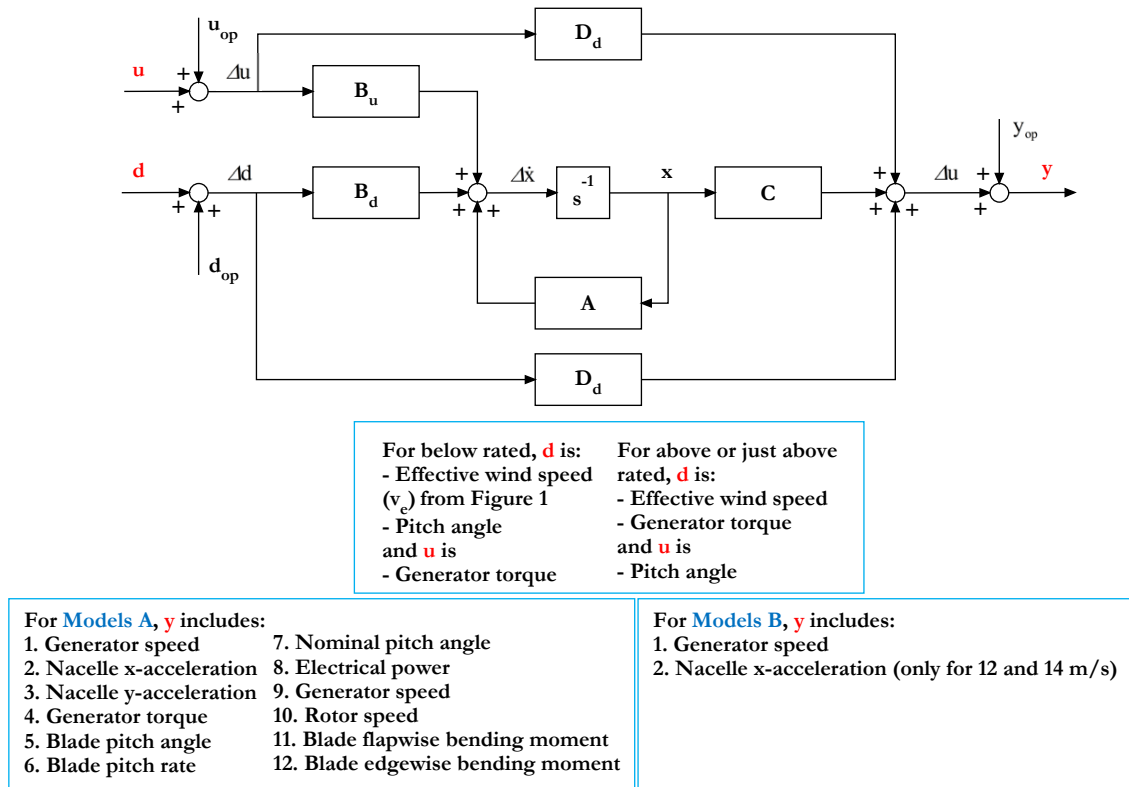


Figure 4. Linearised wind turbine model.

12 and 14 m/s) are defined as

$$\Delta \mathbf{y}(t) = \mathbf{y}(t) - \mathbf{y}_{op}(t) \quad (17)$$

$$\Delta \mathbf{u}_T(t) = \mathbf{u}_T(t) - \mathbf{u}_{T,op}(t) \quad (18)$$

$$\Delta \mathbf{x}(t) = \mathbf{x}(t) - \mathbf{x}_{op}(t) \quad (19)$$

$\mathbf{y}(t)$, $\mathbf{u}_T(t)$, and $\mathbf{x}(t)$ represent the output, input, and states, respectively, and $\mathbf{y}_{op}(t)$, $\mathbf{u}_{T,op}(t)$, and $\mathbf{x}_{op}(t)$ are the operating points around which the models are linearised. The resulting Matlab/SIMULINK simulation model is depicted in Figure 4.

The outputs for the linear models, as shown in the figure, are

1. Measured generator speed
2. Nacelle x-acceleration (also known as tower acceleration)
3. Nacelle y-acceleration
4. Generator torque
5. Blade pitch angle
6. Blade pitch rate
7. Nominal pitch angle
8. Electrical power
9. Generator speed
10. Rotor speed
11. Blade flapwise bending moment
12. Blade edgewise bending moment

The inputs are

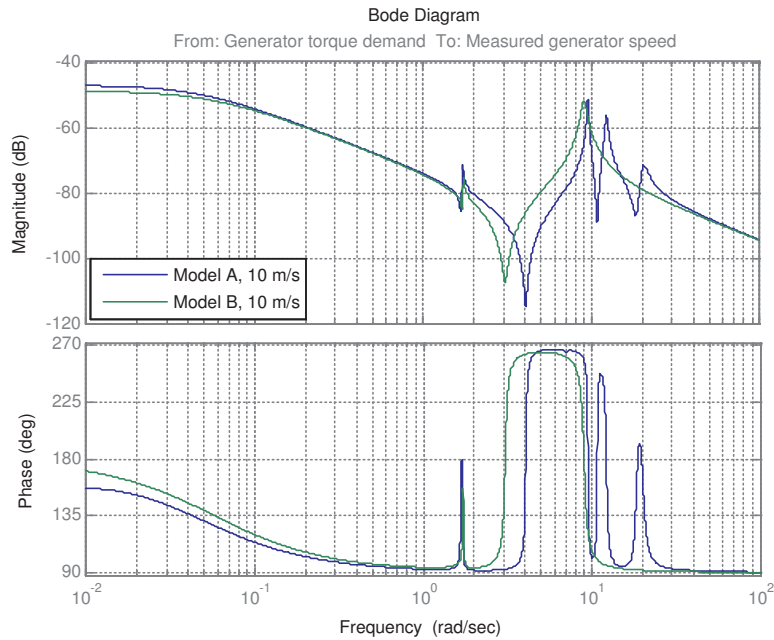


Figure 5. Frequency responses of Model A and Model B at a mean wind speed of 10 m/s.

1. Wind speed
2. Pitch angle
3. Generator torque demand

When designing a generator torque controller for the below rated model, wind speed among the inputs, $\mathbf{u}_T(t)$, is considered to be a disturbance, $\mathbf{d}(t) \in \mathbb{R}^1$, and pitch angle set to zero modifying (16) as follows.

$$\begin{aligned}\Delta \dot{\mathbf{x}}(t) &= A\Delta \mathbf{x}(t) + B_u \Delta u(t) + B_d \Delta \mathbf{d}(t) \\ \Delta \mathbf{y}(t) &= C\Delta \mathbf{x}(t) + D_u \Delta u(t) + D_d \Delta \mathbf{d}(t)\end{aligned}\quad (20)$$

where the input, $u(t) \in \mathbb{R}^1$, is now generator torque (or generator torque demand) only.

When designing pitch controllers for the just above and above rated models, $u(t)$ would be pitch angle only, wind speed is treated as a disturbance, and generator torque is set to a constant value. This is also manifested in Figure 4. Therefore, MPC and LQG controllers utilise the following single-input, single-output (SISO) equations:

$$\begin{aligned}\Delta \dot{\mathbf{x}}(t) &= A\Delta \mathbf{x}(t) + B_u \Delta u(t) \\ \Delta y_g(t) &= C_g \Delta \mathbf{x}(t) + D_{u,g} \Delta u(t)\end{aligned}\quad (21)$$

where $y_g(t) \in \mathbb{R}^1$ denotes generator speed, and $D_{u,g}$ zero.

To wit, the torque (at 10 m/s) and pitch (at 12 and 14 m/s) controllers control generator speed by varying generator torque demand and by active pitching, respectively.

As previously mentioned, these models are referred to as Models A throughout the paper. The open-loop frequency responses of these models for mean wind speeds of 10, 12, and 14 m/s are depicted in Figures 5 and 6, in comparison to the open-loop frequency responses of the models introduced in the following section (i.e. Models B).

2.3. Wind Turbine Linear Models B (linearisation via symbolic differentiation)

The nonlinear model equations provided in [20] are linearised via symbolic differentiation in this section. The parameters of the same turbine as the one used to derive Models A (i.e. Supergen 5MW

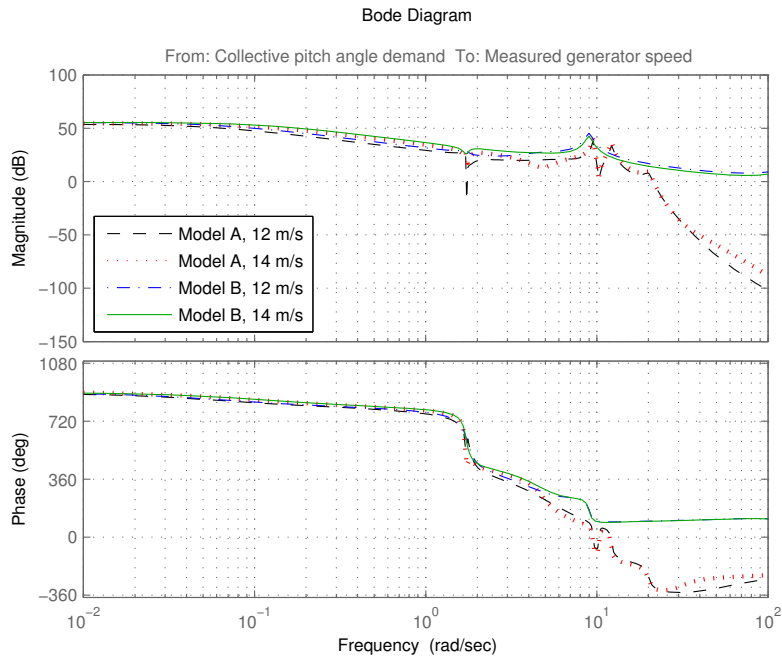


Figure 6. Frequency responses of Models A and Models B at mean wind speeds of 12 and 14 m/s.

exemplar turbine) are exploited. Being only 12th order for at 10 m/s and 11th order at 12 and 14 m/s, they are much simpler models than Models A. As depicted in Figure 4, in below rated wind speed (i.e., at 10 m/s) the process input is torque demand, while in above rated wind speed (i.e., at 12 and 14 m/s) the process input is pitch demand. Hence, it is a plausible outcome that the orders of the models are different. These models are also in state space form, and referred to as Models B throughout the paper, as previously mentioned. The open-loop frequency responses for 10, 12, and 14 m/s are depicted in Figures 5 and 6, in comparison to the open-loop frequency responses of Models A from Section 2.2.

The inputs are same as Models A, but the outputs are

1. Measured generator speed for 10 m/s, 12 m/s, and 14 m/s
2. Nacelle x-acceleration (also known as tower acceleration) for 12 and 14 m/s.

3. MODEL PREDICTIVE CONTROL

MPC is briefly revised in this section and designed based on both Models A and Models B to investigate the controllers' dependence on the choice of linear models used during the design process. As mentioned in Section 1, applying both of the controllers to a neutral model that is neither Models A nor Models B would provide a fair comparison, but such a model is not available for this study. Applying the Models A and B-based controllers to Models A and B, respectively, would be the normal approach when designing the controllers but would not provide the comparison. Instead, the Models A-based controllers are applied to Models A, and the Models B-based controllers to also Models A. Since both controllers are applied to Models A, which are chosen over Models B due to being higher order, only the Models B-based controllers unfairly experience process-model mismatch. These experiments are to show that despite process-model mismatch, the Models B-based controllers outperform the Models A-based controllers, demonstrating importance of the choice of the design models. Moreover, the differences between these models provide a degree of process-model mismatch to test the robustness of design.

3.1. Controller Description

For the following state-space model, which can be obtained by discretising the continuous model in (21),

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B_u u_k \quad (22)$$

$$y_{k+1} = C_g \mathbf{x}_{k+1} \quad (23)$$

the prediction equations for MPC can be derived as [26] (note that the D state-space matrix is zero since no direct feedthrough is allowed in MPC)

$$\underbrace{\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{x}_{k+2} \\ \mathbf{x}_{k+3} \\ \vdots \\ \mathbf{x}_{k+n_y} \end{bmatrix}}_{\mathbf{x}_{\rightarrow}} = \underbrace{\begin{bmatrix} A \\ A^2 \\ A^3 \\ \vdots \\ A^{n_y} \end{bmatrix}}_{P_{xx}} \mathbf{x}_k + \underbrace{\begin{bmatrix} B_u & 0 & 0 & \dots \\ AB_u & B_u & 0 & \dots \\ A^2 B_u & AB_u & B_u & \dots \\ \vdots & \vdots & \vdots & \vdots \\ A^{n_y-1} B_u & A^{n_y-2} B_u & A^{n_y-3} B_u & \dots \end{bmatrix}}_{H_{xx}} \underbrace{\begin{bmatrix} u_{k+1} \\ u_{k+2} \\ u_{k+3} \\ \vdots \\ u_{k+n_y} \end{bmatrix}}_{u_{\rightarrow}} \quad (24)$$

and

$$\underbrace{\begin{bmatrix} y_{k+1} \\ y_{k+2} \\ y_{k+3} \\ \vdots \\ y_{k+n_y} \end{bmatrix}}_{y_{\rightarrow}} = \underbrace{\begin{bmatrix} C_g A \\ C_g A^2 \\ C_g A^3 \\ \vdots \\ C_g A^{n_y} \end{bmatrix}}_P \mathbf{x}_k + \underbrace{\begin{bmatrix} C_g B & 0 & 0 & \dots \\ C_g AB & C_g B & 0 & \dots \\ C_g A^2 B & C_g AB & C_g B & \dots \\ \vdots & \vdots & \vdots & \vdots \\ C_g A^{n_y-1} B & C_g A^{n_y-2} B & C_g A^{n_y-3} B & \dots \end{bmatrix}}_H u_{\rightarrow} \quad (25)$$

where n_y denotes prediction horizon, and u_{\rightarrow} can be rewritten as

$$\underbrace{\begin{bmatrix} u_{k+1} \\ u_{k+2} \\ \vdots \\ u_{k+n_u-1} \\ u_{k+n_u} \\ u_{k+n_u} \\ \vdots \\ u_{k+n_u} \end{bmatrix}}_{u_{\rightarrow}} \quad (26)$$

MPC requires the prediction horizon, n_y , not to be smaller than the control horizon, n_u ; that is, $n_u \leq n_y$.

The control solution can be attained with minimising the following objective function: [27]

$$J = \left\| r - \mathbf{H} u_{\rightarrow} - \mathbf{P} \hat{x}_k - \mathbf{L} d \right\|_2^2 + \lambda \left\| u_{\rightarrow} \right\|_2^2 \quad (27)$$

subject to the following constraints

$$\underline{u}_i \leq u_i \leq \bar{u}_i \quad (28)$$

$$\Delta \underline{u}_i \leq \Delta u_i \leq \Delta \bar{u}_i \quad (29)$$

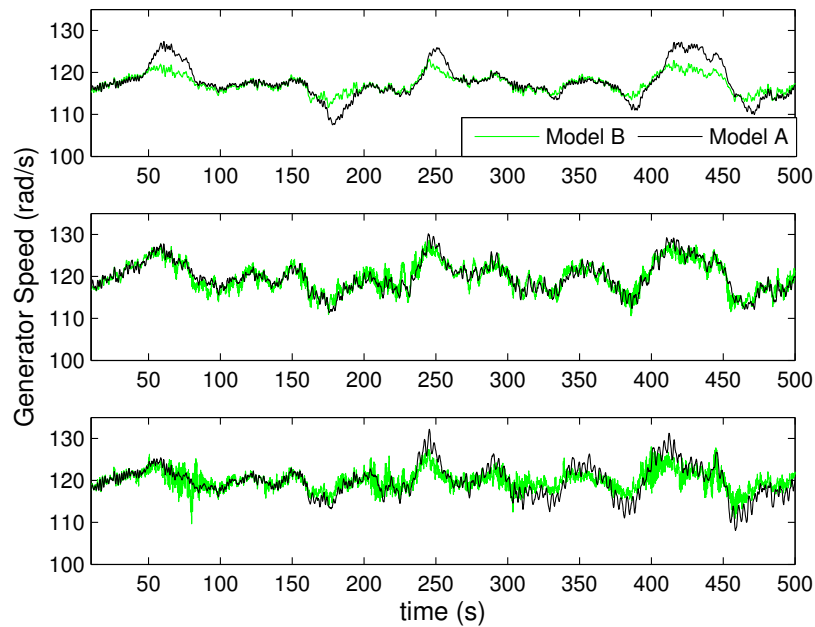


Figure 7. MPC based on Models A and Models B; generator speed (control output); time responses for mean wind speeds of 10 m/s, 12 m/s, and 14 m/s.

where \bar{u}_i denotes the upper limit on u_i , \underline{u}_i the lower limit, r the reference signal, \mathbf{L} a vector of ones, whose size is simply dependent on the prediction horizon, and Δu_i the rate of change of input. The offset $d = y - \hat{y}$ is included to give unbiased predictions and offset correction. The first $\|\cdot\|_2^2$ term is to reduce the reference tracking error and the second $\|\cdot\|_2^2$ term to reduce the control action. Therefore, λ provides a trade-off between these two conflicting problems. \hat{x}_k comes from the internal model here, but a state estimator such as the Kalman filter could also be utilised. For the optimisation, the Matlab function, “quadprog” (the interior-point-convex algorithm) is employed.

3.2. Implementation, Tuning, and Simulation

The tuning parameters can be summarised as follows:

- At a mean wind speed of 10 m/s, for the Models A-based controllers, prediction horizon, n_y , is set to 25, control horizon, n_u , to 25, and control weighting, λ , to 3×10^{-18} . For the Models B-based controllers, prediction horizon, n_y , is set to 30, control horizon, n_u , to 30, and control weighting, λ , to 5.5×10^{-8} .
- At a mean wind speed of 12 m/s, for the Models A-based controllers, prediction horizon, n_y , is set to 25, control horizon, n_u , to 25, and control weighting, λ , to 1×10^{-4} . For the Models B-based controllers, prediction horizon, n_y , is set to 30, control horizon, n_u , to 30, and control weighting, λ , to 1.4×10^{-2} .
- At a mean wind speed of 14 m/s, for the Models A-based controllers, prediction horizon, n_y , is set to 25, control horizon, n_u , to 25, and control weighting, λ , to 1.0×10^{-12} . For the Models B-based controllers, prediction horizon, n_y , is set to 30, control horizon, n_u , to 30, and control weighting, λ , to 6.3×10^2 .

The measured outputs (i.e. generator speed) are depicted in Figure 7. The time responses for the Models B-based controllers at each mean wind speed are satisfactory in terms of fluctuation, remaining well below 12 %, which is often within the controller design specification. High frequency oscillation is observed throughout the simulations, especially at mean wind speeds of 12 and 14 m/s. The bottom subplot, i.e. 14 m/s, is rescaled at around 68 s in Figure 8. It oscillates at

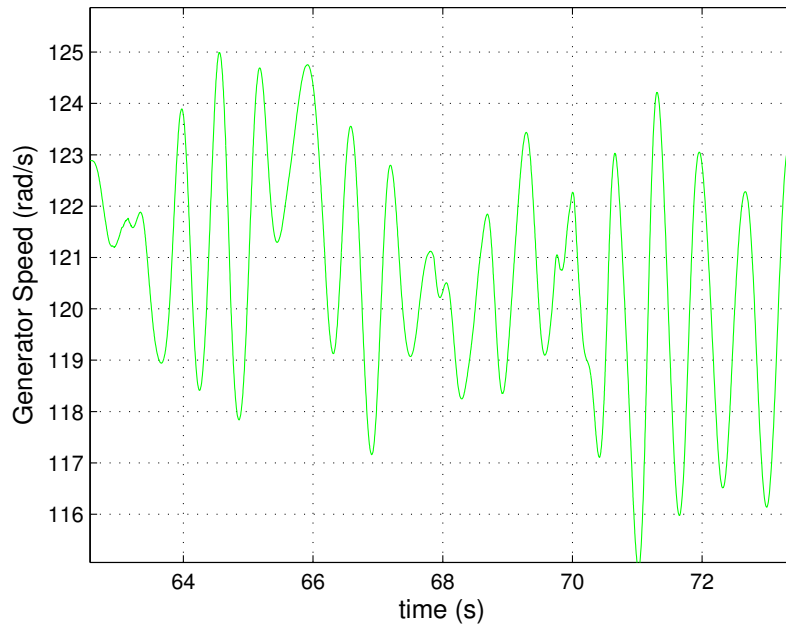


Figure 8. Rescaled plot of Figure 7; MPC based on Models B; generator speed (control output); time responses for mean wind speeds of 14 m/s.

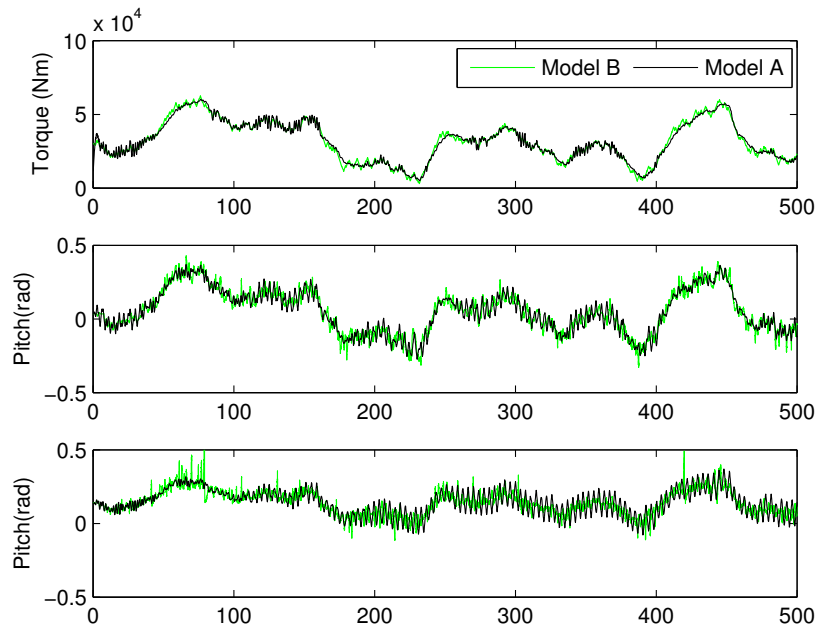


Figure 9. MPC based on Models A and Models B; torque demand (10 m/s) and pitch (12 and 14 m/s) (control input); time responses for mean wind speeds of 10 m/s, 12 m/s, and 14 m/s.

frequency of approximately 10 rad/s, corresponding to the drive-train mode, which can be removed through the use of a drive-train damper [5]. This is explained in more detail with the corresponding power spectra in Figures 10 and 11 below.

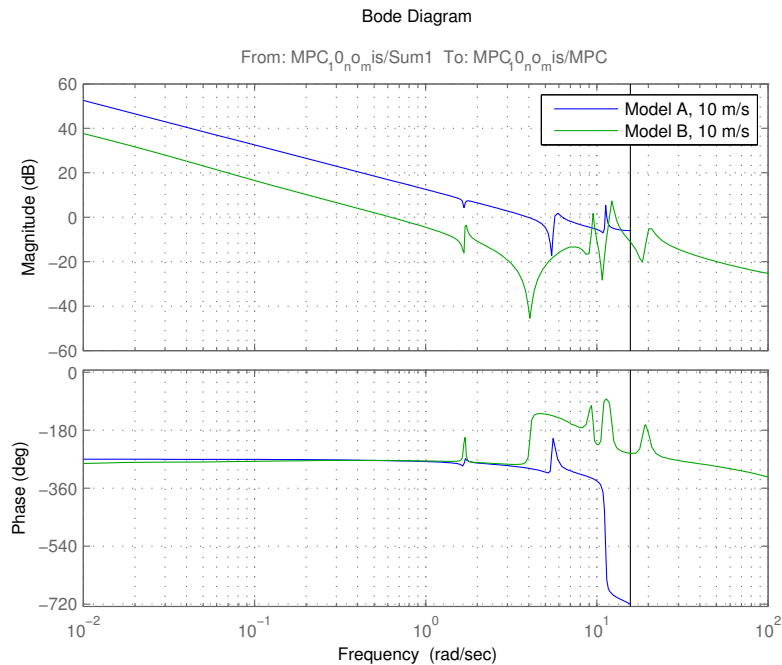


Figure 10. MPC based on Models A and Models B for a mean wind speed of 10 m/s; open loop frequency responses; $T_s=0.2$ s (sampling time) for Models A and $T_s=0.02$ s for Models B.

Although the fluctuations remain well below 12 % also for the Models A-based controllers, sustained oscillation (at frequency of around 5.5 rad/s) can be observed at mean wind speeds of 12 and 14 m/s. It would result in increased loads on the rotor that would propagate down the power-train and impact on the drive train components, e.g. gearbox and shaft. The cause is explained with the frequency response in Figure 11.

Torque demand and pitch angle, i.e. the control inputs, are also presented in Figure 9. As mentioned previously, the turbine operates in mode 3 at 10 m/s and in mode 4 at 12 and 14 m/s. Generator speed is controlled by varying torque in mode 3 and by pitching in mode 4. Fluctuation on the control inputs is not as limited as generator speed, the control output. Pitch angle ranges from -5° to 35° for the turbine considered in this paper.

The open-loop frequency responses for the Models A-based controllers demonstrate poorer results in comparison to the Models B-based controllers as depicted in Figures 10 and 11. The open-loop system of a controller in this paper is referred to as the product of the process (i.e. the turbine model) and the controller (open-loop). With information obtained from the open-loop responses [21], such as gain crossover frequency, phase margin, etc, the response of the close-loop system, such as the size of the control action, sensitivity to uncertainty, stability, etc, can be speculated.

The Models A-based controllers are more sensitive to sampling time, T_s , and 0.2 s needs to be chosen to improve the results. The vertical lines that appear at 20 rad/s in the figures are due to discretisation with this sampling time. The Models B-based controllers are less sensitive to sampling time, and a smaller sampling time, i.e. 0.02 s, could successfully be exploited.

At a mean wind speed of 10 m/s, Figure 10 depicts that the gain crossover frequencies of the Model A and B-based controllers are approximately 4 and 0.6 rad/s, respectively. Note that Control System Toolbox™ in Matlab is exploited here for producing open-loop frequency responses in bode plots throughout the paper. Due to the characteristics of wind, the controllers should be tuned to give a gain crossover frequency in the range of 0.6 to 2 rad/s [28]. Gain crossover frequencies over this range may lead to large control action, hence actuator saturation, especially in high wind speeds. Thus, the Models A-based controller is too aggressive. Gain crossover frequencies below this range

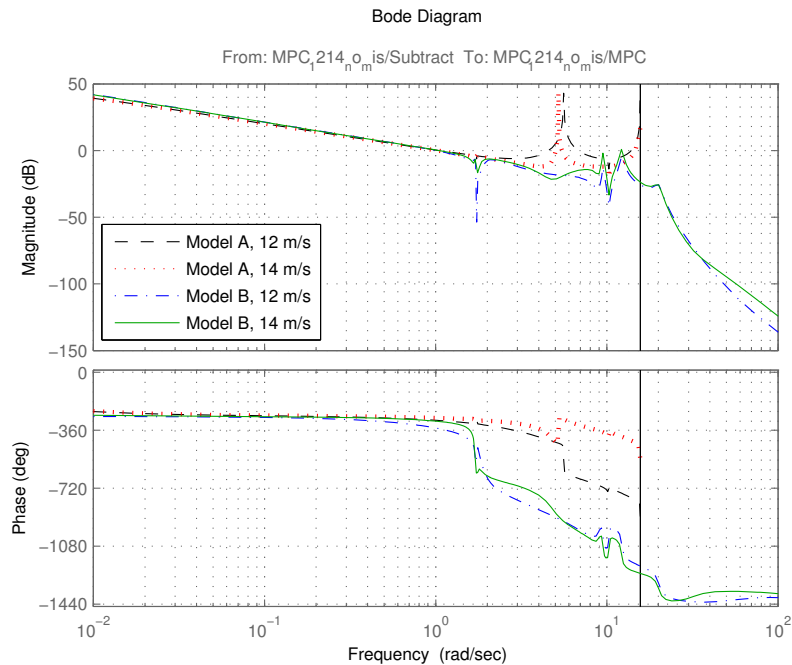


Figure 11. MPC based on Models A and Models B at mean wind speeds of 12 and 14 m/s; open loop frequency responses; $T_s=0.2$ s (sampling time) for Models A and $T_s=0.02$ s for Models B.

could lead to too slow control action. Phase margins for the Model A and B-based controllers are respectively some 90° and 75° , indicating that their closed-loop responses would be stable – note that the MPC controllers incorporate a positive feedback.

For mean wind speeds of 12 and 14 m/s, Figure 11 illustrates that the gain crossover frequencies are approximately 1 rad/s for both controllers. However, the peaks from the Models A-based controllers remain mostly above 0 dB slightly, indicating that the controllers would be sensitive to uncertainty and noise. The peaks at around 5.5 rad/s in particular cause the sustained oscillation that is observed in the time response depicted in Figure 7. The controllers should also be tuned to ensure that the peaks at high frequencies are kept as small as possible. The responses of the Models B-based controllers demonstrate significantly improved results keeping the peaks at 5.5 rad/s below 0 dB, hence no sustained oscillation at the specific frequency is visible in the time response. However, the peaks crossing over 0 dB slightly at around 10 rad/s cause the oscillations at 10 rad/s as briefly discussed above with Figures 7 and 8. As mentioned previously, this peak is normally removed by subsequently applying a drive-train damper, but it is not discussed in this paper.

Phase margins are around 61° (12 m/s) and 78° (14 m/s) for the Models A-based controllers and 12° (12 m/s) and 60° (14 m/s) for the Models B-based controllers, indicating that their closed-loop responses would be stable.

3.3. Summary

Both the Models A and B-based controllers are applied to Models A. Models A are chosen due to being higher order than Models B. Only the Models B-based controllers thus experience process-model mismatch. Nonetheless, the results presented in this section demonstrate that the Models A-based controllers are more aggressive at a mean wind speed of 10 m/s and significantly more sensitive to uncertainty and noise and more aggressive than the Models B-based controllers at mean wind speeds of 12 and 14 m/s. The peaks crossing over 0 dB in the open-loop frequency response in Figure 11 cause sustained oscillation at a specific frequency as demonstrated in the time response in Figure 7. Models B are significantly simpler models than Models A, and Models A have a tendency

to cause the controllers to become over-aggressive, i.e. active at higher frequencies, and to lack robustness. It is thus recommended that the control design model be kept as simple as possible when designing MPC controllers.

4. LINEAR QUADRATIC GAUSSIAN CONTROL

LQG is composed of a Kalman filter and a Linear Quadratic Regulator (LQR) [29, 30]. The Kalman filter estimates the states, and LQR controls the process using these state estimates. LQG is briefly revised in this section and designed based on both Models A and Models B to investigate the controllers' dependence on the choice of linear models used during the design process as with the MPC controllers introduced in Section 3.

The Models A-based controllers are applied to Models A, and the Models B-based controllers also to Models A. Consequently, only the Models B-based controllers experience process-model mismatch. This test is to investigate if the Models B-based controllers, being designed based on a simpler model, would outperform the Models A-based controllers despite the disadvantage (i.e., process-model mismatch), equally to designing the MPC controllers in Section 3. Moreover, the process-model mismatch allows the robustness of design to be tested.

4.1. Controller Description

4.1.1. *Kalman Filter* Given the following continuous model from (20) with additional noise terms

$$\begin{aligned}\dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + B_u u(t) + w(t) \\ y_g(t) &= C_g \mathbf{x}(t) + D_{u,g} u(t) + v(t)\end{aligned}\quad (30)$$

where $u(t)$ denotes the input, $w(t)$ white process noise, and $v(t)$ measurement noise, the following assumptions could be made:

$$E(w(t)) = E(v(t)) = 0 \quad (31)$$

$$E(w(t)w^T(t)) = Q_N \quad (32)$$

$$E(v(t)v^T(t)) = R_N \quad (33)$$

$$E(w(t)v^T(t)) = N_N \quad (34)$$

The state estimate, $\hat{\mathbf{x}}(t)$, that minimises the following steady-state error covariance can be constructed [31].

$$P = \lim_{t \rightarrow \infty} E((\mathbf{x}(t) - \hat{\mathbf{x}}^T(t))(\mathbf{x}(t) - \hat{\mathbf{x}}^T(t))) \quad (35)$$

The optimal solution is the Kalman filter with the following equation:

$$\dot{\hat{\mathbf{x}}}(t) = A\hat{\mathbf{x}}(t) + B_u u(t) + K(\hat{y}(t) - y(t)) \quad (36)$$

The Kalman filter gain, K , is derived by solving an algebraic Riccati equation as follows: [32]

$$K = (PC^T + (B_d(Q_N D_d^T + N_N)))(R_N + D_d N_N + N_N^T D_d^T D_d Q_N D_d^T)^{-1} \quad (37)$$

The estimator utilises the inputs, $u(t)$, and the measured output, $y(t)$, to generate the output estimates, $\hat{y}(t)$, and state estimates, $\hat{\mathbf{x}}(t)$. $K(\hat{y}(t) - y(t))$ in (36) continuously corrects $\hat{y}(t)$ to cause the estimator to track $y(t)$ more closely.

4.1.2. *Linear Quadratic Regulator* LQR solves the following optimisation problem offline:

$$J_{lq} = \int_0^{\infty} \hat{\mathbf{x}}^T(t) Q \hat{\mathbf{x}}(t) + u^T(t) R u(t) dt \quad (38)$$

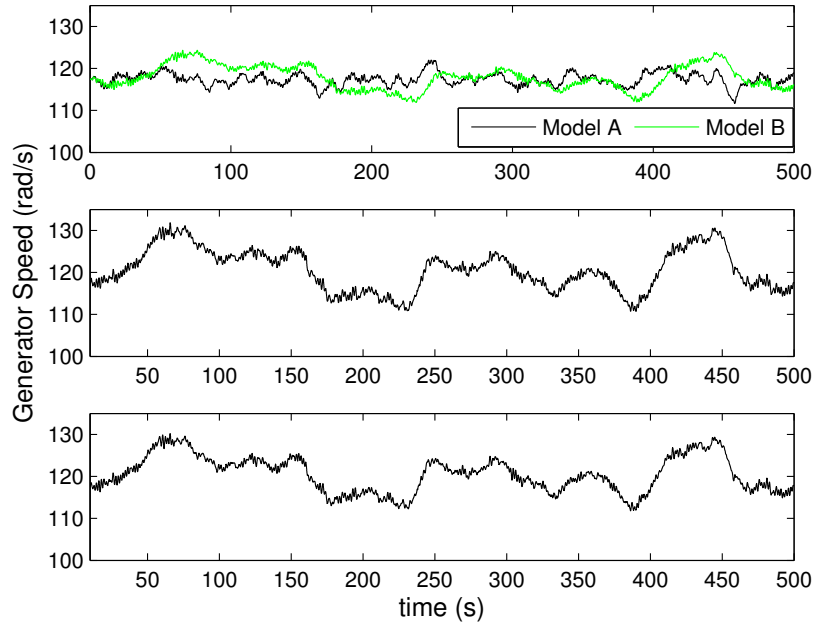


Figure 12. LQG based on Models A and Models B; time responses at mean wind speeds of 10, 12, and 14 m/s; the controllers based on Models B become unstable at mean wind speeds of 12 and 14 m/s, hence no plots are included.

where Q is an $r \times r$ symmetric positive-definite matrix and R an $m \times m$ symmetric positive-definite matrix. The first term is to reduce the reference tracking error and the second term is to reduce the control action.

The optimal state-feedback LQR controller for minimising the performance index, J_{lq} , in (38) is a simple matrix gain of the following form:

$$u(t) = -L\hat{x}(t) \quad (39)$$

where \hat{x} is the state estimates from the Kalman filter, and L is a $m \times n$ matrix as follows:

$$L = (D_u^T Q D_u + R)^{-1} (B^T P + D_u^T Q C) \quad (40)$$

P is the unique positive-definite solution to the following algebraic Riccati Equation:

$$A^T P + P A + C^T Q C - (P B + C^T Q D_u) (D_u^T Q D_u + R)^{-1} (B^T P + D_u^T Q C) = 0 \quad (41)$$

4.2. Implementation, Tuning, and Simulation

Tuning parameters for the LQG controllers are as follows:

- At a mean wind speed of 10 m/s, for the Model A-based controllers, Q is set to the identity matrix I and R to 2×10^{-7} . For the Model B-based controllers, Q is set to I , and R to 7×10^{-8} .
- At a mean wind speed of 12 m/s, for the Models A-based controllers, Q is set to $C_g^T C_g$ and R set to 2×10^2 . For the Models B-based controllers, Q is initially set to $C_g^T C_g$, $Q(10, 10)$ subsequently modified to $Q(10, 10) \times 10^5$, and R to 5×10^2 .
- At a mean wind speed of 14 m/s, for the Models A-based controllers, Q is initially set to $C_g^T C_g$ and R to 4×10^2 . For the Models B-based controllers, Q is initially set to $C_g^T C_g$, $Q(10, 10)$ subsequently modified to $Q(10, 10) \times 10^5$, and R set to 6×10^2 .

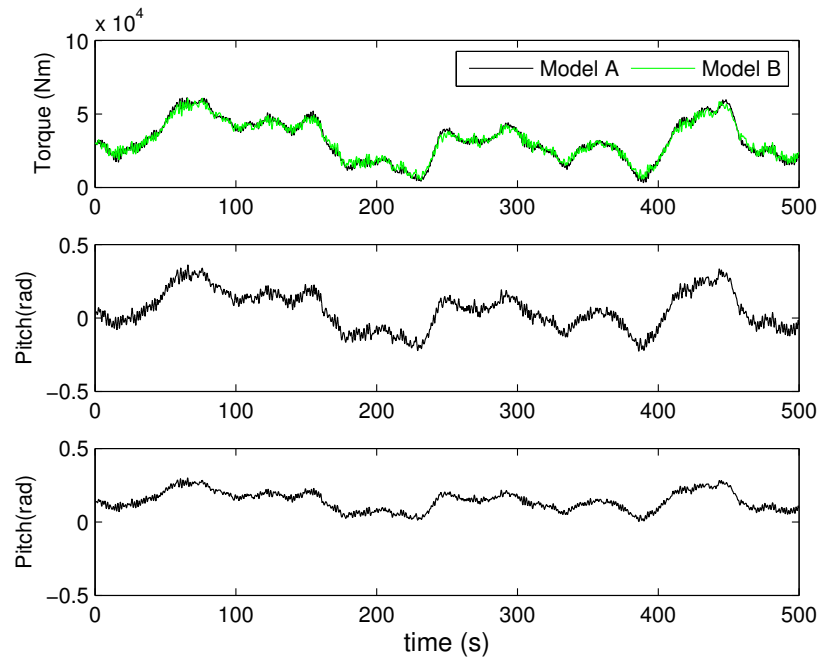


Figure 13. LQG based on Models A and Models B; torque demand (10 m/s) and pitch (12 and 14 m/s) (control input); time responses for mean wind speeds of 10 m/s, 12 m/s, and 14 m/s.

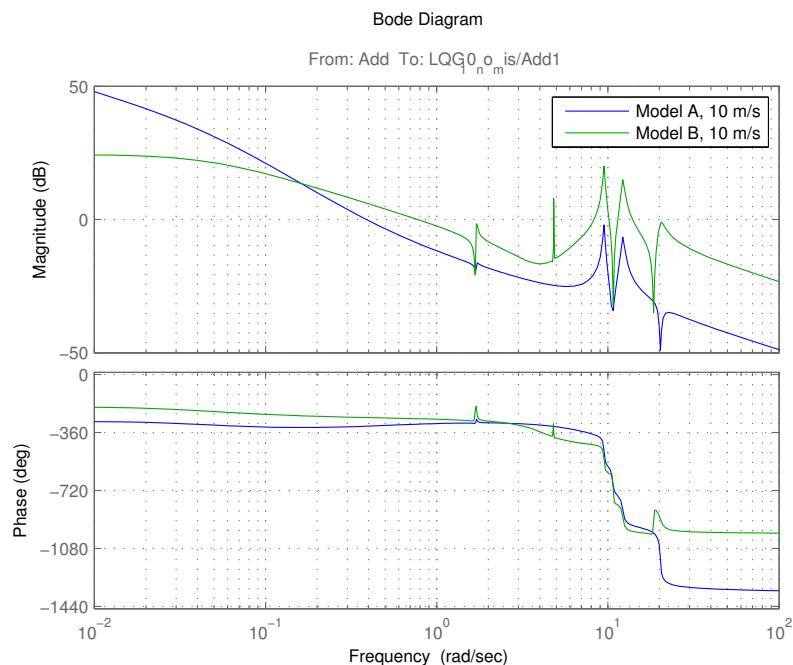


Figure 14. LQG based on Models A and Models B at a mean wind speed of 10 m/s; open loop frequency responses.

Figure 12 demonstrates that the time responses at a mean wind speed of 10 m/ produced by both the Models A and Models B-based controllers are satisfactory. At mean wind speeds of 12 and 14 m/s, only the Models A-based controllers perform satisfactorily, and the Models B-based

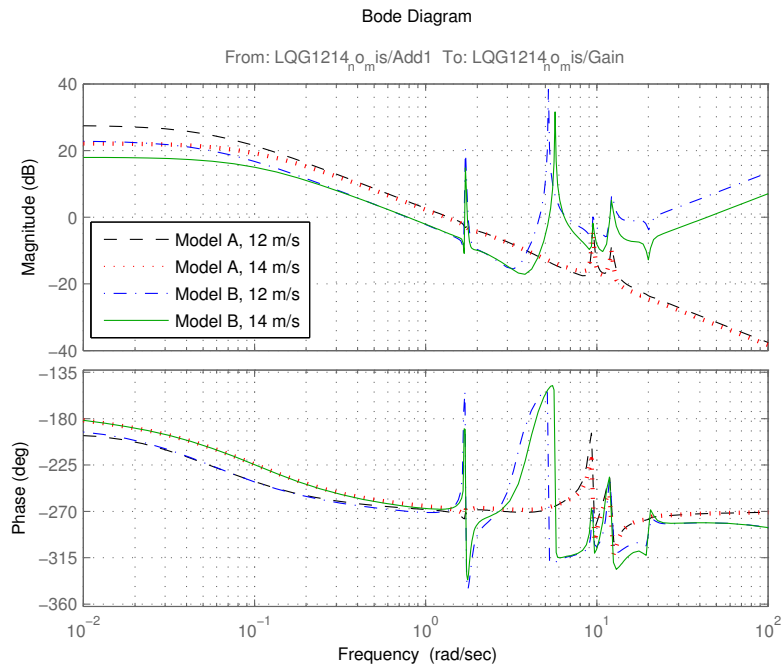


Figure 15. LQG based on Models A and Models B at mean wind speeds of 12 and 14 m/s; open loop frequency responses.

controllers become unstable – hence no plots are displayed for the Models B-based controllers. Note, however, that the Models B-based controllers provide satisfactory results when applied to Models B, meaning that the Models A-based controllers do not necessarily outperform Models B-based controllers although the results are not included here. The corresponding control action, i.e. torque at 10 m/s and pitch at 12 and 14 m/s, is depicted in Figure 13.

The open-loop frequency response depicted in Figure 14 suggests that the Model A-based controller would be too slow having a low gain crossover frequency, i.e. less than 0.4 rad/s, while the gain crossover frequency of the Models B-based controllers is within the range – it was explained in Section 3 that the gain crossover frequency should be in the range of 0.6 to 2 rad/s due to the characteristics of wind. It was also previously explained that the peaks crossing over 0 dB at around 10 rad/s, corresponding to the drive-train mode, can be removed by subsequently applying a drive-train damper. Phase margins for the Model A and Model B-based controllers are around 49° and 92° , respectively, indicating that their closed-loop responses would be stable – note that LQG controllers incorporate a positive feedback.

For mean wind speeds of 12 and 14 m/s, Figure 15 illustrates that the gain crossover frequencies for the Models A and B-based controllers are ideally 1.3 and 0.7 rad/s, respectively. The peaks produced by the Model B-based controllers in Figure 15 cross over 0 dB, and the frequency responses do not roll off at higher frequencies, indicating that the controllers would be sensitive to uncertainty and noise. The time responses in Figure 13 agree with these frequency responses; that is, the Model B-based controllers become unstable (hence, no plots for the Model B-based controllers are included in Figure 12), while the Model A-based controllers produce satisfactory time responses, as anticipated from the frequency responses. Phase margins are around 90° for both the controllers, indicating that their closed-loop responses would be stable.

4.3. Summary

Both the Models A and B-based controllers are applied to the Models A-based controllers. Models A are chosen as they are higher order, hence being closer to the nonlinear model in complexity. As

a result, only the Models B-based controllers experience process-model mismatch. The simulations carried out in this section are to investigate if the Models B-based controllers, being based on simpler models, would provide superior results to the Models A-based controllers as with the MPC controllers designed in Section 3.

The Models A-based controllers perform satisfactorily, having no process-model mismatch, while the Models B-based controllers do not cope with process-model mismatch and fail to produce satisfactory results, not to mention improved results, especially at mean wind speeds of 12 and 14 m/s. Nonetheless, it cannot be concluded that the Models A-based controllers outperform the Models B-based controllers since only the Models B-based controllers experience process-model mismatch and they perform fine when applied to Models B, although the results are not presented here.

The LQG controllers are difficult and time-consuming to tune, and it has a tendency to be over-aggressive, i.e., active at higher frequencies, and lacks robustness. Their properties are less dependent on the choice of linear model used during the design process in comparison to the MPC controllers.

5. EXTENSION TO LINEAR QUADRATIC GAUSSIAN

In Section 4, the controllers designed based on Models B are aggressive and unstable for 12 and 14 m/s. In this section, the controllers are stabilised by improving the robustness.

The LQG controller is a model-based controller that includes an internal model. The Kalman filter included in the controller design acts as the internal model (with “Kalman gain”) providing the controller (i.e., Linear Quadratic Regulator) with the state estimation. The Kalman filter assumes that the disturbance model can be identified online or given in advance. When the disturbance model can be neither identified online nor given in advance, or when any other uncertainty, which includes unknown disturbance, modelling error, and faults, emerges, the performance of the Kalman filter, and hence the LQG controller, could degrade significantly.

In this section, the Kalman filter is replaced by an observer whose gain is obtained to minimise the effect of uncertainty and disturbances, thereby improving the robustness of the controller. The resulting controllers are referred to as “observer-based LQR controllers” throughout this paper.

As mentioned previously, the observer design is based on the well-known Luenberger observer and could be applied to any other control algorithms that utilise an estimator or an internal model such as Inverse Model Control (IMC), MPC, etc. It would also be interesting to discover how other types of observers, such as Unknown Input Observer (UIO) [33], would improve the original LQG controller in future.

5.1. Observer Design

The mathematical description of the observer is as follows:

$$\begin{aligned}\dot{\tilde{\mathbf{x}}}(t) &= A\tilde{\mathbf{x}}(t) + B_u u(t) + K(y_g(t) - \tilde{y}_g(t)) \\ \tilde{y}_g(t) &= C_g \tilde{\mathbf{x}}(t)\end{aligned}\quad (42)$$

where $y_g(t)$ and $\tilde{y}_g(t)$ denote the measurement of generator speed and the estimate, respectively, and $u(t)$ represents the control action. The term involving the observer gain, K , should correct the observer estimate continuously to cause $\tilde{y}_1(t)$ to follow $y_1(t)$ more closely. This implies that the effects of process-model mismatch and disturbances can be reduced by optimising K . Derivation of an optimal gain, K is described here.

5.1.1. Observer Gain in Frequency Domain Process-model mismatch and disturbances may be described by additional terms, $\tilde{\mathbf{d}}_1(t) \in \mathbb{R}^r$ and $\tilde{d}_2(t) \in \mathbb{R}^1$, and (20) can be modified as follows:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + B_u u(t) + \tilde{\mathbf{d}}_1(t) \\ y_g(t) &= C_g \mathbf{x}(t) + \tilde{d}_2(t)\end{aligned}\quad (43)$$

Subtracting $\dot{\tilde{\mathbf{x}}}(t)$ in (42) from $\dot{\mathbf{x}}(t)$ in (43), the equation for the residual $r(t) = y_g(t) - \tilde{y}_g(t)$ can be derived as follows: [33]

$$\begin{aligned}\dot{\mathbf{e}}(t) &= (A - KC)\mathbf{e}(t) + \tilde{\mathbf{d}}_1(t) - K\tilde{d}_2(t) \\ r(t) &= C\mathbf{e}(t) + \tilde{d}_2(t)\end{aligned}\quad (44)$$

where

$$\dot{\mathbf{e}}(t) = \dot{\mathbf{x}}(t) - \dot{\tilde{\mathbf{x}}}(t) \quad (45)$$

Taking the Laplace transform of (44), the following equation can be obtained:

$$r(s) = C(sI - A + KC)^{-1}\tilde{\mathbf{d}}_1(s) + (I - KC(sI - A + KC)^{-1})\tilde{d}_2(s) \quad (46)$$

Therefore, the effects of process-model mismatch and disturbances can be minimised using the following performance indices:

$$J_1(K) = \|C(sI - A + KC)^{-1}\|_\infty \quad (47)$$

$$J_2(K) = \|I - C(sI - A + KC)^{-1}K\|_\infty \quad (48)$$

where $\|\cdot\|_\infty$ denotes L_∞ norm, also known as infinity norm. By minimising $J_1(K)$ and $J_2(K)$, the maximums of the largest singular values of $C(sI - A + KC)^{-1}$ and $I - KC(sI - A + KC)^{-1}$, which correspond to the peak gains of the frequency response, are minimised. Hence, the effects of process-model mismatch and disturbances can be minimised.

The problem now is to find K such that $J_1(K)$ and $J_2(K)$ are minimised. However, it is likely that K causes instability. This can be prevented by parameterising K via the Eigenstructure Assignment (EA) method [33] summarised as follows.

Note that, for improved results, the vectors $\tilde{\mathbf{d}}_1(t)$ and $\tilde{d}_2(t)$ may be reconstructed to approximate the effect of disturbances and process-model mismatch. If some bounds on these signals can be estimated then good robustness may be incorporated in the design by adding these as constraints to the minimisation problems of the performance indices. However, this approach is not pursued here but good performance is still achieved by carefully tuning the controller.

5.1.2. Parameterisation via Eigenstructure Assignment Method When conducting an optimisation to minimise $J_1(K)$ and $J_2(K)$ from (47) and (48), it is important to ensure that the stability of the observer is always guaranteed when no disturbance or fault is present, and this leads to more complex constrained optimisation problem. To guarantee the stability condition, the use of the EA method, which parameterises K , is suggested. The method has an advantage of allowing the eigenvalues in predefined regions and is summarised as follows. It is assumed that the eigenvalues are always real for the sake of brevity. Since the observer design problem is the ‘‘dual problem’’ of the controller design, \mathbf{v}_i is the i_{th} eigenvector of $A^T - C^T K^T$ corresponding to the i_{th} eigenvalue λ_i as follows:

$$(A^T - C^T K^T)\mathbf{v}_i = \lambda_i \mathbf{v}_i \quad (49)$$

$$\mathbf{v}_i = -(\lambda_i I - A^T)^{-1} C^T w_i \quad (50)$$

where $w_i = K^T \mathbf{v}_i$. There are now two design parameters, w_i and λ_i instead of one design parameter K . These design parameters still do not guarantee the stability of the observer.

The eigenvalue λ_i , one of the design parameters, is generally not required to be placed at a specific point in the s or z -planes but rather in a predefined region to satisfy the stability condition. This in turn provides more relaxed design freedom as follows:

$$\lambda_i \in [L_i, U_i] \quad (51)$$

where L_i and U_i ($i = 1, \dots, n$), respectively, denote the upper and lower bounds. By defining an equation for the eigenvalue as

$$\lambda_i = L_i + (U_i - L_i \sin^2(z_i)) \quad (52)$$

$z_i \in \mathbb{R}$ ($i = 1, \dots, n$) becomes a design parameter instead of λ_i . Any z_i subsequently guarantees the stability condition.

Finally, the two design parameter vectors W and Z have been defined and the performance indices in (47) and (48) can be rewritten as follows:

$$J_1(W, Z) = \|C(sI - A + KC)^{-1}\|_\infty \quad (53)$$

$$J_2(W, Z) = \|I - C(sI - A + KC)^{-1}K\|_\infty \quad (54)$$

where

$$K = [WV^{-1}]^T \quad (55)$$

Having redefined the optimisation problems as finding Z and W from finding K only, the stability condition is always guaranteed.

Since, two performance indices (53) and (54) need to be minimised simultaneously, a multi-objective optimization technique is required. The use of an evolutionary algorithm that exploits the genetic algorithm is proposed here to solve the multi-objective optimization problem. An evolutionary algorithm, or more specifically, a genetic algorithm, is used here for solving this multi-objective optimisation problem. It computes for an optimisation solution by performing a parallel search of the solution space or the objective function as opposed to the calculation of cost function gradients. An advantage is that it is relatively straightforward to implement. Several analytical methods for solving this multi-objective optimisation problem have also been proposed, which include the optimisation via singular value decomposition [34] and the optimisation via optimal projection [35], but we opt for the multi-objective based on genetic algorithm. It would be interesting to see how this multi-objective optimisation performs against the analytical methods, but the comparison is beyond the scope of this paper.

A genetic algorithm can be combined with the method of inequalities and moving boundaries method to set up a multi-objective optimisation. For the genetic algorithm part, the Genetic Algorithm toolboxTM in Matlab is exploited. The method of inequalities and moving boundaries method are utilised to provide the genetic algorithm toolbox with an appropriate objective function. The details of the algorithm is not presented in this paper, but readers could refer to [36, 37], instead.

5.2. Implementation, Tuning, and Simulation

In Section 4, the LQG controllers fail to produce satisfactory results at mean wind speeds of 12 and 14 m/s. The observer-based LQR controllers introduced in this section, to improve the controllers' robustness, are designed based on Models B and applied to Models A for 12 and 14 m/s. The differences between these models provide a degree of process-model mismatch to test the robustness of design.

Tuning parameters for the LQR controllers are Q and R in (41). The parameters are set as follows:

- At a mean wind speed of 12 m/s, Q is initially set to $9 \times 10^2 I$, $Q(4, 4)$ is subsequently modified to $Q(4, 4) \times 10^7$ and R is set to 2×10^8 .
- At a mean wind speed of 14 m/s, Q is initially set to $9 \times 10^2 I$, $Q(7, 7)$ is subsequently modified to $Q(7, 7) \times 10^7$ and R is set to 2×10^8 .

The open-loop frequency responses for 12 and 14 m/s are depicted in Figures 16 and 17 (in blue) in comparison to the LQG controllers (green) that fail to stabilise in Section 4. Figure 16 depicts that, for 12 m/s, the observer-based LQR controllers would be less sensitive to uncertainty and noise because all the peaks remain below 0 dB and the response rolls off at high frequencies. The gain crossover frequency remaining just below 1 rad/s indicates that the controllers would be

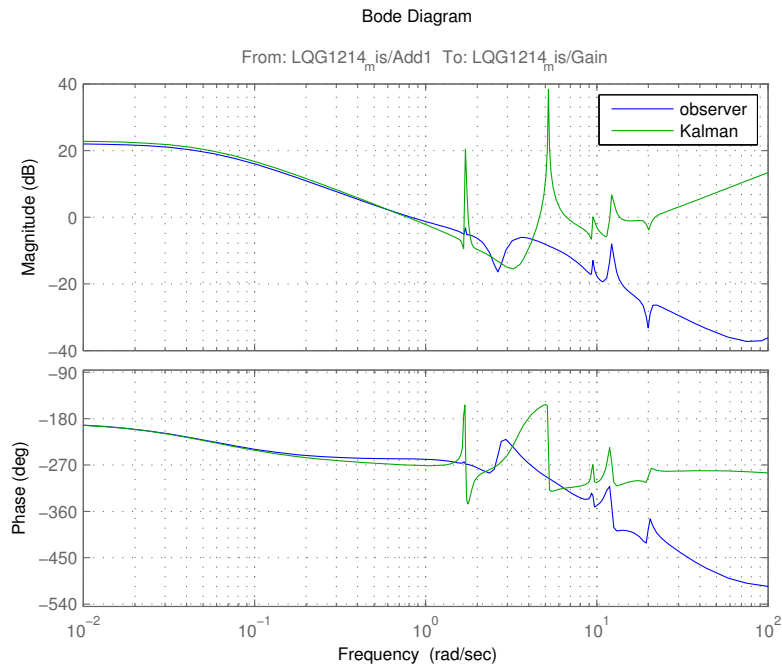


Figure 16. observer-based LQR based on Models B at a mean wind speed of 12 m/s; open loop frequency responses.

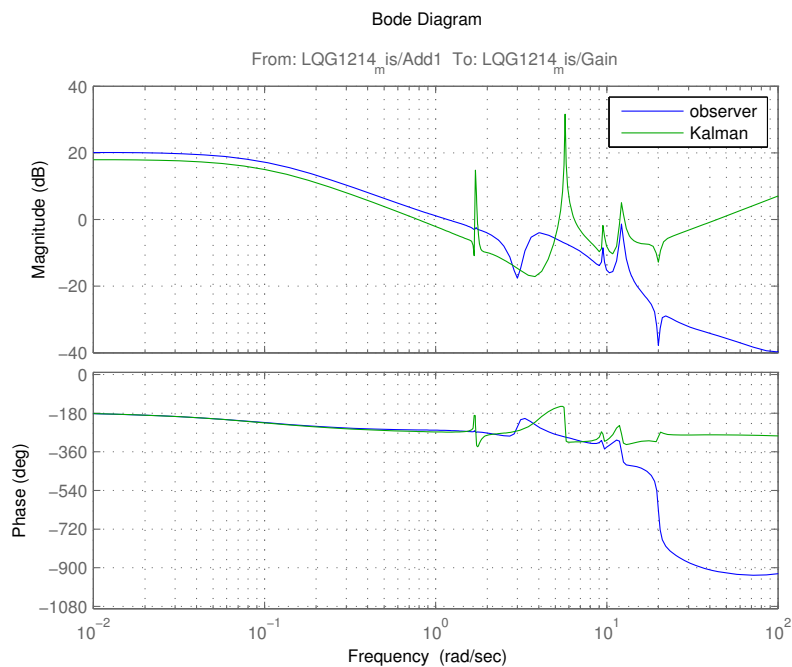


Figure 17. observer-based LQR using Models B at a mean wind speed of 14 m/s; open loop frequency responses.

neither too aggressive nor too slow. Moreover, phase margins are around 95° (12 m/s) and 90° (14 m/s), indicating that their closed-loop responses would be stable. Figure 17 demonstrates similar results, and, therefore, the same conclusions could be drawn for 14 m/s. The measured outputs (i.e.,

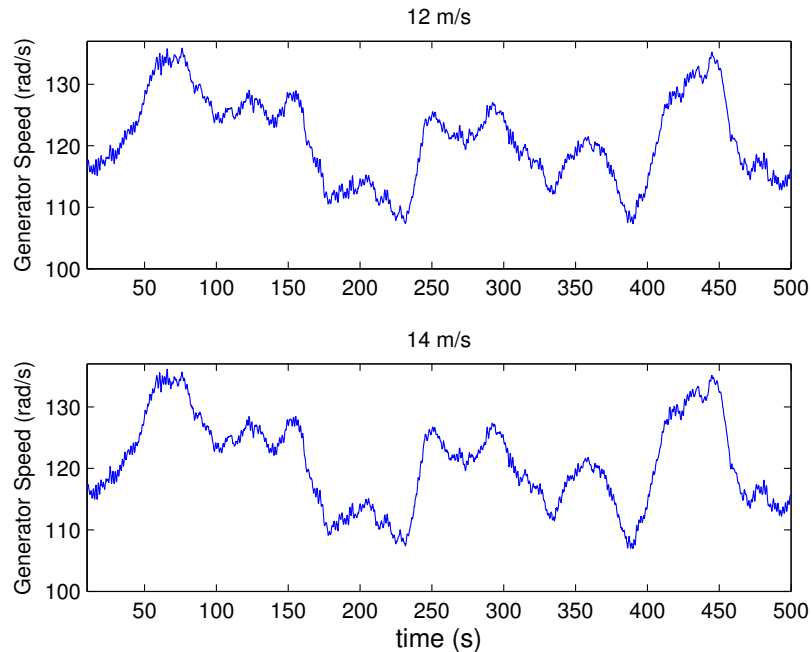


Figure 18. observer-based LQR using Models B; generator speed (control output); time responses at mean wind speeds of 12 and 14 m/s.

generator speed) are depicted in Figure 18. In contrast to the performance of the original LQG controllers depicted in Figure 12, the time responses of the observer-based LQR controllers for both 12 and 14 m/s are satisfactory as the fluctuations remain below 12 %.

5.3. Summary

The original LQG controllers based on Models B produce poor results, particularly for mean wind speeds of 12 and 14 m/s in Section 4. In this section, improved results are obtained by enhancing the robustness of the controllers. It is achieved by replacing the Kalman filter with an observer whose gain is obtained to minimise the effect of uncertainty and disturbance. The results demonstrate significant improvement achieved by the observer-based LQR controllers. The methodology could be applicable to any control algorithms that utilise a state estimator or internal model, such as MPC, Inverse Model Control, etc.. Furthermore, it is exploited in [33] in a similar approach to design a fault monitoring system. Moreover, the extra computation is performed offline, i.e. during the initialisation stage, and only takes a few seconds on an Intel® Core™i7-4790K @4.00 GHz machine.

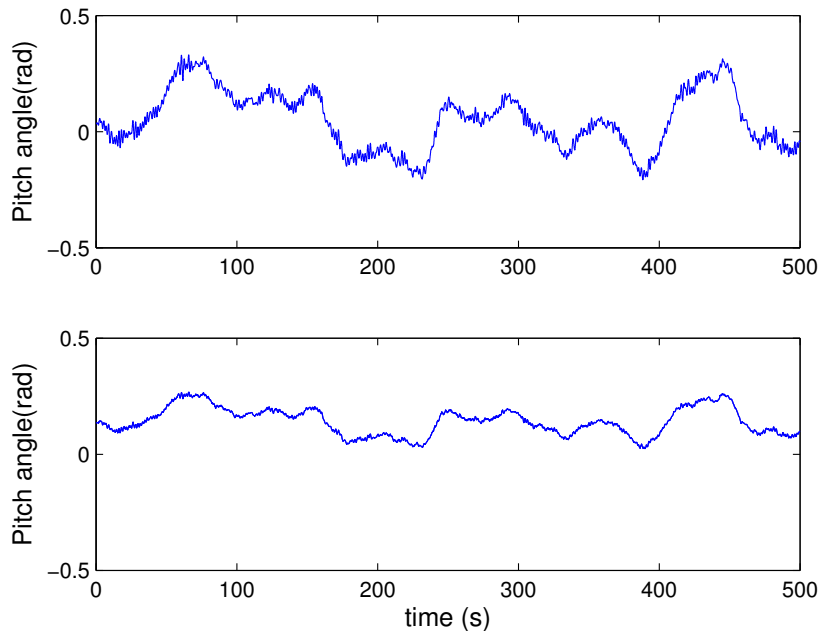


Figure 19. observer-based LQR using Models B; pitch (control input); time responses at mean wind speeds of 12 and 14 m/s.

6. CONCLUSIONS

Two linear dynamic models (including wind speed models) of the Supergen 5MW exemplar turbine are constructed for three operating points, below rated wind speed (10 m/s), just above rated wind speed (12 m/s) and above rated wind speed (14 m/s). One is obtained in Bladed using its linearisation toolbox, and the other using the standard linearisation method (i.e., via symbolic differentiation). In order to investigate the MPC and LQG controllers' dependence on the choice of linear model used during the design process, the controllers designed based on Models A and B. The performance is tested by applying both the controllers, i.e., Models A-based and Models B-based controllers, to Models A. Models A are chosen since no other model is available and Models A are higher order, thereby being closer to the nonlinear higher order model (from which the linear models are derived) in complexity.

When designing MPC controllers, since only the Models B-based controllers experience process-model mismatch, the Models A-based controllers are expected to provide improved results. However, Models B-based controllers outperform Models A-based controller despite the handicap. Also, the differences between these models provide a degree of process-model mismatch to test the robustness of design. Models B, being simpler models, are less aggressive and more robust to uncertainty and noise than Models B-based controllers. It is therefore recommended that the control design models be kept as simple as possible when designing MPC controllers for wind turbines.

The LQG controller is difficult and time-consuming to tune than the MPC controllers. Its properties are less dependent on the choice of linear model used during the design process in comparison to the MPC controllers. The controller tuned based on Models B demonstrate poorer results but it is due to process-model mismatch. The application of the Models B-based controllers to Models A show that the controller has a tendency to be over-aggressive, i.e., active at higher frequencies, and lacks robustness.

The last part of this paper improves the LQG controller in order to improve its robustness, especially at higher frequencies. The Kalman filter included in the original LQG design is replaced

by an observer whose gain is designed to minimise the effects of uncertainty and noise. The results demonstrate significant improvement, and both the frequency and time responses produce improved results in comparison to the original LQG design. The extra computation required for the observer design is performed offline and only takes a few seconds on an Intel® Core™i7-4790K @4.00 GHz machine.

ACKNOWLEDGEMENT

The authors wish to acknowledge the support of the EPSRC for the Supergen Wind Energy Technologies Consortium, grant number EP/H018662/1.

REFERENCES

1. Luo N, Vidal Y, Acho L. *Wind Turbine Control and Monitoring*. Springer, 2014.
2. Burton T, Sharpe D, Jenkins N, Bossanyi E. *Wind Energy Handbook*. John Wiley & Sons, Ltd, 2001.
3. Leithead WE, Connor B. Control of variable speed wind turbines: design task. *International Journal of Control* 2000; **13**:1189–1212.
4. Garcia-Sanz M, Houpis CH. *Wind Energy Systems: Control Engineering Design*. CRC Press, 2012.
5. Chatzopoulos A. Full Envelope Wind Turbine Controller Design for Power Regulation and Tower Load Reduction. PhD Thesis, University of Strathclyde 2011.
6. Bianchi FD, Battista HD, Mantz RJ. *Wind Turbine Control Systems: Principles, Modelling and Gain Scheduling Design*. Springer, 2006.
7. Camacho EF, Bordons C. *Model Predictive Control in the Process Industry (Advances in Industrial Control)*. Springer, 2011.
8. Grancharova AI, Johansen TA. *Explicit Nonlinear Model Predictive Control: Theory and Applications*. Springer, 2012.
9. Grimble MJ, Johnson MA. *Optimal control and stochastic estimation: theory and applications*. Wiley, 1988.
10. trentelman H, Stoorvogel AA, Hautus M. *Control Theory for Linear Systems (Communications and Control Engineering)*. Springer, 2012.
11. Skogestad S, Postlethwaite I. *Multivariable Feedback Control: Analysis and Design*. 2 edn., Wiley, 2005.
12. Brosilow C, Joseph B. *Techniques of Model-based Control*. Prentice Hall Professional, 2002.
13. Körber A. Extreme and Fatigue Load Reducing Control for Wind Turbines: A Model Predictive Control Approach using Robust State Constraints. PhD Thesis, Technische Universität Berlin 2014.
14. Schlipf D, Schlipf DJ, Kühn M. Nonlinear model predictive control of wind turbines using LIDAR. *Wind Energy* 2013; **16**:1107–1129.
15. Riboldi C. Advanced Control Laws for Variable-Speed Wind Turbines and Supporting Enabling Technologies. PhD Thesis, Politecnico di Milano 2012.
16. Henriksen L, Hansen M, Poulsen N. Wind turbine control with constraint handling: a model predictive control approach. *IET Control Theory and Applications* 2012; **6** (11):1722–1734.
17. Kumar AA, Stol K. Scheduled Model Predictive Control of a Wind Turbine. *Proceedings of the European Wind Energy Conference, Warsaw*, 2010.
18. Østergaard KZ. Robust, Gain-Scheduled Control of Wind Turbines. PhD Thesis, Aalborg University, Denmark 2008.
19. *Bladed User Manual v4.3*.
20. Leithead W, Connor B. Control of variable speed wind turbines: Dynamic models. *International Journal of Control* 2000; **73**: **13**:1173 – 1188.
21. Ogata K. *Modern Control Engineering*. Fifth edn., Pearson Education International, 2009.
22. Dorf RC, Bishop RH. *Modern Control Systems*. Eleventh edn., Pearson Prentice Hall, 2008.
23. Leithead WE. Effective wind speed models for simple wind turbine simulations. *Proceedings of 14th British Wind Energy Association (BWEA) Conference, Nottingham*, 1992.
24. Neilson VW. Individual Blade Control for Fatigue Load Reduction of Large-scaled Wind Turbines: Theory and Modelling. Master's Thesis, Department of Electronic and Electrical Engineering, University of Strathclyde 2010.
25. Munteanu I, Bratcu AI, Cutululis NA, Ceangă E. *Optimal Control of Wind Energy Systems: Towards a Global Approach*. Springer, 2007.
26. Rossiter JA. *Model-Based Predictive Control: a Practical Approach*. CRC Press, 2005.
27. Maciejowski JM. *Predictive Control with Constraints*. Prentice Hall, 2000.
28. Leith DJ, Leithead WE. Appropriate realization of gain-scheduled controllers with application to wind turbine regulation. *International Journal of Control* 1996; **65**(2).
29. Clements DJ, Anderson BDO. *Singular Optimal Control: The Linear-Quadratic Problem*. Springer-Verlag, 1978.
30. Hespanha JP. Undergraduate Lecture Notes on LQG/LQR Controller Design. University of California Santa Barbara, Electrical and Computer Engineering.
31. Franklin GF, Powell JD, Workman WL. *Digital Control of Dynamic Systems*. Second edn., Addison-Wesley, 1990.
32. Lewis F. *Optimal Estimation*. John Wiley & Sons, Inc, 1986.
33. Chen J, Patton RJ. *Robust model-based fault diagnosis for dynamic systems*. Kluwer Academic Publishers, 1999.
34. Gwak KW, Masada GY. Structural Analysis and Optimization of Nonlinear Control Systems Using Singular Value Decomposition. *Journal of Dynamic Systems, Measurement, and Control* 2004; **127**(1):105–113.

35. Chen YL, Ke YL. Multi-objective VAR planning for large-scale power systems using projection-based two-layer simulated annealing algorithms. *IEE Proceedings of Generation, Transmission and Distribution*, vol. 151, 2004; 555 – 560.
36. Elsayed S, Sarker R, Essam D. Improved genetic algorithm for constrained optimization. *2011 International Conference on Computer Engineering & Systems (ICCES)*, 2011.
37. Konak A, Coit DW, Smith AE. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering and System Safety* 2006; **91**:992–1007.