Mechatronics 21 (2011) 645–659

Contents lists available at ScienceDirect

Mechatronics

journal homepage: www.elsevier.com/locate/mechatronics

Robust and fault-tolerant linear parameter-varying control of wind turbines

Christoffer Sloth [a,*,1], Thomas Esbensen [b], Jakob Stoustrup [b]

[a] Department of Computer Science, Aalborg University, DK-9220 Aalborg East, Denmark
[b] Automation and Control, Department of Electronic Systems, Aalborg University, DK-9220 Aalborg East, Denmark

ARTICLE INFO

Article history:
Available online 5 March 2011

Keywords:
Wind turbine control
Linear parameter-varying (LPV) control
Fault-tolerant control
Robust control

ABSTRACT

High performance and reliability are required for wind turbines to be competitive within the energy market. To capture their nonlinear behavior, wind turbines are often modeled using parameter-varying models. In this paper we design and compare multiple linear parameter-varying (LPV) controllers, designed using a proposed method that allows the inclusion of both faults and uncertainties in the LPV controller design. We specifically consider a 4.8 MW, variable-speed, variable-pitch wind turbine model with a fault in the pitch system.

We propose the design of a nominal controller (NC), handling the parameter variations along the nominal operating trajectory caused by nonlinear aerodynamics. To accommodate the fault in the pitch system, an active fault-tolerant controller (AFTC) and a passive fault-tolerant controller (PFTC) are designed. In addition to the nominal LPV controller, we also propose a robust controller (RC). This controller is able to take into account model uncertainties in the aerodynamic model.

The controllers are based on output feedback and are scheduled on an estimated wind speed to manage the parameter-varying nature of the model. Furthermore, the AFTC relies on information from a fault diagnosis system.

The optimization problems involved in designing the PFTC and RC are based on solving bilinear matrix inequalities (BMIs) instead of linear matrix inequalities (LMIs) due to unmeasured parameter variations. Consequently, they are more difficult to solve. The paper presents a procedure, where the BMIs are rewritten into two necessary LMI conditions, which are solved using a two-step procedure.

Simulation results show the performance of the LPV controllers to be superior to that of a reference controller designed based on classical principles.

1. Introduction

Motivated by environmental concerns and the depletion of fossil fuels, increasing attention is being paid to wind energy, which is one of the most promising sustainable energy sources. From a control point of view, a wind turbine is a challenging machine, since it is driven by a stochastic input, which is poorly known. A modern wind turbine is controlled not only to maximize power production, but also to reduce loads, minimize acoustic noise emissions, and meet power quality grid codes.

Wind turbines inherently exhibit nonlinear dynamics, motivating the use of advanced control techniques such as gain-scheduled control to continuously adapt to the dynamics of the plant. Since many wind turbines are installed at remote locations, the introduction of fault-tolerant control is considered a suitable way of improving reliability of wind turbines and lowering costs of repairs. Finally, the lack of accurate models must be countered by robust control strategies capable of securing stability and satisfactory performance despite model uncertainties, see [1].

In this paper a three-bladed horizontal-axis, variable-speed, variable-pitch wind turbine is considered. The aerodynamic properties of the wind turbine are functions of the pitch angles of the blades, the speed of the rotor, and the wind speed. The wind exerts torque and thrust on the rotor. The aerodynamic torque is transferred to the generator through a drive train, which upscales the rotational speed of the rotor, and the aerodynamic thrust is transferred to the tower-top.

In terms of control, the wind turbine operates in two distinct regions, illustrated in Fig. 1. At low wind speeds, in the partial load region, the turbine is controlled to maximize the power output. This is achieved by adjusting the generator torque to obtain an optimum ratio between the tip speed of the blades and the wind speed. At higher wind speeds, in the full load region, the wind turbine is controlled to reduce loads by producing a rated power output at a constant rotor speed, which is obtained by pitching the blades to adjust the efficiency of the rotor, while applying a constant generator torque. In this paper only operation in the full load region is considered.

* Corresponding author.
E-mail address: csloth@cs.aau.dk (C. Sloth).
[1] Supported by MT-LAB, a VKR Centre of Excellence.

0957-4158/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mechatronics.2011.02.001

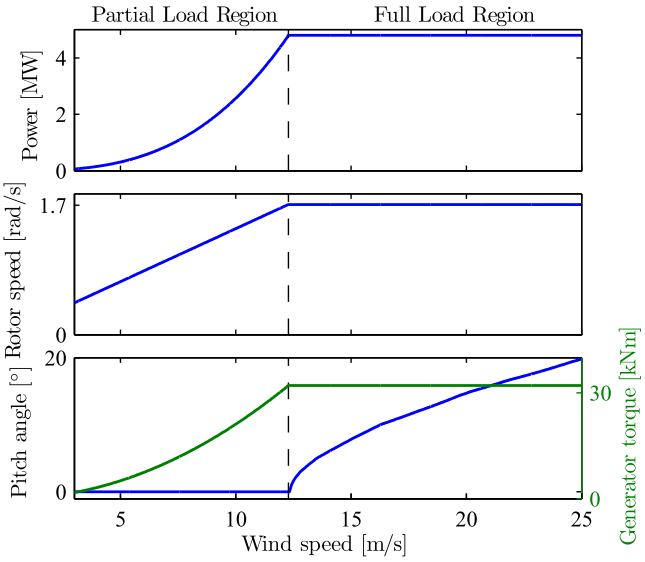

Fig. 1. Power curve, rotor speed trajectory, and overview of control signals for the partial and full load regions.

Due to the varying dynamic behavior of wind turbines along their nominal operating trajectory, wind turbine controllers typically consist of multiple gain-scheduled controllers, which are designed to operate in the proximity of a certain operating point. In [2] a gain-scheduling approach for classical controllers is presented and in [3,4] a controller for the whole operating range is designed by introducing bumpless transfer between robust controllers. The underlying assumption for such control schemes is that the parameters only change slowly compared to the system dynamics, which is generally not satisfied. Additionally, classic gain-scheduling controllers only ensure performance guarantees and stability at the operating points where the linear controllers are designed. A more comprehensive overview of wind turbine control is found in [5].

A systematic way of designing controllers for systems with linearized dynamics that vary significantly with the operating point is within the framework of linear parameter-varying (LPV) control. Here, a controller is synthesized to satisfy a performance specification for all possible parameter values within a specified model and for the specified rate of variation of the parameters. An LPV controller can be synthesized after solving an optimization problem subject to linear matrix inequalities (LMIs).

In previous work, LPV controllers have been developed for wind turbines considering a nominal plant model [1,6]. However, in control systems for wind turbines, robustness and fault-tolerance capabilities are important properties which should be considered in the design process, calling for a generic and powerful tool to manage parameter-variations and model uncertainties. In this paper, in addition to design procedures for nominal controllers for parameter-varying models, also design procedures for controllers, which obtain robustness and active/passive fault-tolerance, are provided. To emphasize the contribution of this paper, a controller is designed, synthesized, and simulated for each partial result; hence, we design the four following controllers. Notice that the names in parentheses are abbreviations used throughout this paper.

1. *Nominal controller (NC)*: LPV controller scheduled to handle the parameter variations along the nominal operating trajectory.
2. *Active fault-tolerant controller (AFTC)*: LPV controller scheduled to handle the parameter variations along the nominal operating trajectory and scheduled to accommodate the fault in the pitch system.
3. *Passive fault-tolerant controller (PFTC)*: LPV controller scheduled to handle the parameter variations along the nominal operating trajectory and resilient towards the fault in the pitch system.
4. *Robust controller (RC)*: LPV controller scheduled to handle the parameter variations along the nominal operating trajectory and robust towards expected variations in the parameters of the aerodynamic model.

The roles of the four controllers are to demonstrate the feasibility of the LPV method when managing parameter variations, robustness, and fault-tolerant control. Indeed, handling known parameter-dependencies, unknown parameter variations, and faults, constitute the main challenges for the application of wind turbine control. The controllers presented in this paper are therefore serious candidates for solving a majority of practical wind turbine control problems, provided a sufficiently good model of the wind turbine is available. Realizing such a controller in practice can sometimes be difficult and may lead to numerical challenges, but taking the precautions described in this paper, we believe that such a controller can also be implemented in practice.

The focus on fault-tolerant control in this paper includes an AFTC [7] and a PFTC [8], both providing a complete solution to the design problem for the considered fault scenario. The intention of the paper is to show feasibility of both solutions, compare the two approaches in terms of design complexity and performance, and allow the best-suited method to be selected for a given design problem. The difference between an AFTC and a PFTC is that an active fault-tolerant controller relies on a fault diagnosis system, which should feed information about the faults to the controller. This knowledge makes it possible for the AFTC to reconfigure according to the current state of the system, but it also introduces some detection time and a risk of false positive and false negative diagnosis, e.g. due to model limitations. A PFTC is optimized for the fault-free situation, while satisfying some degraded performance requirements in the fault scenario. The degradation of performance requirements is what separates reliable controllers from robust controllers, as robust controllers have the same performance guarantee throughout the entire parameter space.

The list of faults occurring in wind turbines is extensive, reflecting the complexity of the machines. On system level, faults occur in sensors, actuators, and system components, ranging from slow gradual faults to abrupt component failures. The occurrence of faults may change the system behavior dramatically. This motivates us to develop methods for fault diagnosis and fault-tolerant control, offering several benefits:

- Prevent catastrophic failures and faults from deteriorating other parts of the wind turbine, by early fault detection and accommodation.
- Reduce maintenance costs by providing remote diagnostic details and avoiding replacement of functional parts, by applying condition-based maintenance instead of time-based maintenance.
- Increase energy production when a fault has occurred by means of fault-tolerant control.

This paper addresses the simple case of a single fault: altered dynamics of the hydraulic pitch system due to low hydraulic pressure. The fault is a gradual fault affecting the control actions of the turbine. The method used also applies to fast parameter variations, i.e. abrupt faults in the extreme case, see [9]. The fault modeling and the motivation for considering this fault originate from [10]. For a more comprehensive treatment of multiple fault types related to wind turbine operation see [9].

This paper is organized as follows: Section 2 describes the wind turbine plant model and the considered fault. In Section 3 a refer-

ence controller based on classical methods is presented to establish a frame of reference for the designed LPV controllers. In Section 4 the optimization problems and controller synthesis procedures are presented for designing the nominal LPV controller and the controllers including the fault-tolerant and robust extensions. Section 5 contains the simulation results and compares the performance of the LPV controllers to the performance of the reference controller. Section 6 concludes the paper.

2. Wind turbine model

A nonlinear wind turbine model is used for simulation of the proposed control algorithms. The model consists of sub-models for: static aerodynamics, the tower, the drive train, the generator, the pitch system and the converter actuator, and the measurement noise. The sub-models are separately explained in the following subsections and combined at the end of this section, and the model parameters are listed in Appendix A. Throughout the sequel, subscript 'r' will refer to rotor parts and subscript 'g' will refer to generator parts.

2.1. Aerodynamic model

The rotor of the wind turbine converts kinetic energy of the wind to rotational energy of the rotor blades and shaft, rotating at the speed $\omega_r(t)$. The power in the wind depends on the wind speed, $v_r(t)$, the air density, ρ, and the swept area, A. From the available power in the swept area, the power transferred to the rotor is given based on the power coefficient, $C_p(\lambda(t),\beta(t))$, which is a function of the pitch angle of the blades, $\beta(t)$, and the ratio between the speed of the blade tip and the wind speed, denoted the tip-speed ratio, $\lambda(t)$. The aerodynamic torque applied to the rotor is given as:

$$T_a(t) = \frac{1}{2\omega_r(t)}\rho A v_r^3(t) C_p(\lambda(t),\beta(t)) \quad [\mathrm{Nm}] \tag{1}$$

The coefficient C_p describes the aerodynamic efficiency of the rotor by the mapping illustrated in Fig. 2.

The thrust exerted by the wind on the rotor, $F_t(t)$, is calculated as shown in (2), where $C_t(\lambda(t),\beta(t))$ is the trust coefficient.

$$F_t(t) = \frac{1}{2}\rho A v_r^2(t) C_t(\lambda(t),\beta(t)) \quad [\mathrm{N}] \tag{2}$$

2.2. Drive train model

The drive train model consists of a low-speed shaft and a high-speed shaft having inertias J_r and J_g, and friction coefficients B_r and B_g. The shafts are interconnected by a transmission having gear ratio N_g, combined with torsion stiffness K_{dt}, and torsion damping B_{dt}. This results in a torsion angle, $\theta_\Delta(t)$, and a torque applied to the generator, $T_g(t)$, at a speed $\omega_g(t)$. The model of the drive train is given as:

$$J_r\dot{\omega}_r(t) = T_a(t) + \frac{B_{dt}}{N_g}\omega_g(t) - K_{dt}\theta_\Delta(t) - (B_{dt}+B_r)\omega_r(t) \quad [\mathrm{Nm}] \tag{3a}$$

$$J_g\dot{\omega}_g(t) = \frac{K_{dt}}{N_g}\theta_\Delta(t) + \frac{B_{dt}}{N_g}\omega_r(t) - \left(\frac{B_{dt}}{N_g^2}+B_g\right)\omega_g(t) - T_g(t) \quad [\mathrm{Nm}] \tag{3b}$$

$$\dot{\theta}_\Delta(t) = \omega_r(t) - \frac{1}{N_g}\omega_g(t) \quad [\mathrm{rad/s}] \tag{3c}$$

2.3. Pitch system model including fault model

The considered wind turbine has a hydraulic pitch system which is modeled as a second-order system with a time delay, t_d, and input $\beta_{ref}(t)$. The natural frequency, ω_n, and damping ratio, ζ, specify the dynamics of the model:

$$\ddot{\beta}(t) = -2\zeta\omega_n\dot{\beta}(t) - \omega_n^2\beta(t) + \omega_n^2\beta_{ref}(t-t_d) \quad [^\circ/\mathrm{s}^2] \tag{4}$$

To represent the limitations of the pitch actuators, the model includes constraints on the slew rate and the range of the pitch angle; see Appendix A for specific values.

2.3.1. Fault model

A drop in the hydraulic pressure affects the dynamics of the pitch system by changing the damping ratio and natural frequency from their nominal values ζ_0 and $\omega_{n,0}$ to their values at low pressure ζ_{lp} and $\omega_{n,lp}$, as described in (6). Low hydraulic pressure is characterized as a gradual fault, since it affects control actions of the turbine. Step responses of the pitch system in the normal and fault conditions are illustrated in Fig. 3. The pressure level is modeled as a convex combination of the vertices of the two parameter sets according to [10].

The dynamics of the pitch system is described as a second-order system

$$\ddot{\beta}(t) = -2\zeta(\theta_f)\omega_n(\theta_f)\dot{\beta}(t) - \omega_n^2(\theta_f)\beta(t) + \omega_n^2(\theta_f)\beta_{ref}(t-t_d) \quad [^\circ/\mathrm{s}^2] \tag{5}$$

where

$$\omega_n^2(\theta_f) = (1-\theta_f)\omega_{n,0}^2 + \theta_f\omega_{n,lp}^2 \quad [\mathrm{rad/s}^2] \tag{6a}$$

$$-2\zeta(\theta_f)\omega_n(\theta_f) = -2(1-\theta_f)\zeta_0\omega_{n,0} - 2\theta_f\zeta_{lp}\omega_{n,lp} \quad [\mathrm{rad/s}] \tag{6b}$$

and $\theta_f \in [0,1]$ is an indicator function for the fault with $\theta_f = 0$ and $\theta_f = 1$ corresponding to normal pressure and low pressure, and $\dot{\theta}(t) \in [-30/\mathrm{s}, 30/\mathrm{s}]$. It is assumed that the terms in (5) are changed linearly between the two vertices.

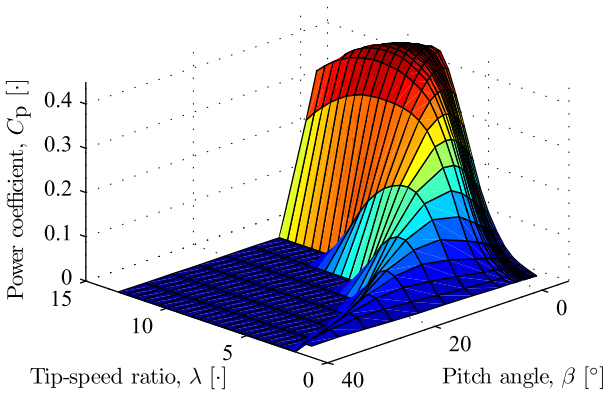

Fig. 2. Illustration of the power coefficient, C_p.

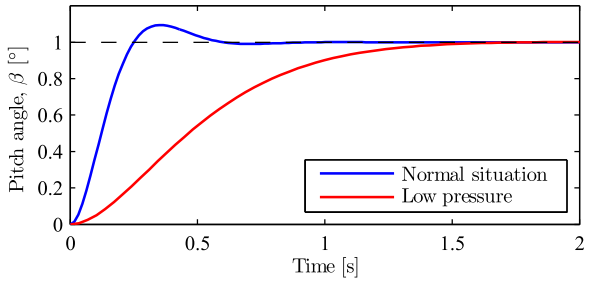

Fig. 3. Step responses of hydraulic pitch model under normal (blue) and fault (red) conditions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

2.4. Generator and converter models

Electric power is generated by the generator, while a power converter interfaces the wind turbine generator output with the utility grid and controls the currents in the generator. The generator torque in (7) is controlled by the reference $T_{g,ref}(t)$. The converter dynamics are approximated by a first-order system with time constant τ_g and time delay $t_{g,d}$. Just as for the model of the pitch system, the slew rate and the operating range of the generator torque are both bounded to match the limitations of the real system.

$$\dot{T}_g(t) = -\frac{1}{\tau_g}T_g(t) + \frac{1}{\tau_g}T_{g,ref}(t - t_{g,d}) \quad [\text{Nm/s}] \tag{7}$$

The power produced by the generator can be approximated from the mechanical power calculated in (8), where η_g denotes the efficiency of the generator, which is assumed constant.

$$P_g(t) = \eta_g \omega_g(t) T_g(t) \quad [\text{W}] \tag{8}$$

2.5. Tower model

Thrust acting on the rotor introduces fore-aft movements of the flexible tower. Sideward movements are ignored in this paper by neglecting yawing and drive train reaction torque on the tower.

The fore-aft movement of the tower is assumed as a linear displacement of the nacelle, acting as a disturbance to the free wind speed, $v_w(t)$. The force acting on the tower at hub height, $F_{th}(t)$, is determined based on the thrust in (2) distributed to the individual blades, assuming stiff blades, stiff tower, and a tower bending moment at the tower base. The tower displacement, $x_t(t)$, is modeled using the spring-damper terminology in (9), including damping coefficient B_t, stiffness coefficient K_t, and mass M_t.

$$M_t\ddot{x}_t(t) = F_{th}(t) - B_t\dot{x}_t(t) - K_t x_t(t) \quad [\text{N}] \tag{9}$$

The effective wind speed at the rotor is described as:

$$v_r(t) = v_w(t) - \dot{x}_t(t) \quad [\text{m/s}] \tag{10}$$

2.6. Assembled model

The interconnection of the wind turbine sub-models is illustrated in Fig. 4. The disturbance input, $v_w(t)$, is provided by a wind model, where tower shadow and wind shear are modeled as in [11] using a turbulence model derived from the wind model in [12]. Furthermore, fore-aft movement of the tower is included in the resulting wind speed as shown in (10).

Available measurements are: generator torque [Nm], generator power [W], pitch angle [°], generator speed [rad/s], and rotor speed [rad/s]; all sampled at a rate of 100 Hz. The measurement noise is modeled as zero-mean white Gaussian noise with the following standard deviations: $\sigma_{Tg} = 90$ Nm, $\sigma_{Pg} = 17$ kW, $\sigma_\beta = 0.2°$, $\sigma_{\omega g} = 0.016$ rad/s, and $\sigma_{\omega r} = 0.025$ rad/s. Considering the gear ratio of 95, the signal-to-noise ratio (SNR) of the measurement of $\omega_r(t)$ is much smaller than the SNR of the measurement of $\omega_g(t)$. This is typical to wind turbines, where the generator speed measurement has a much higher precision, whereas the rotor speed measurement, if available, will only be used in a diagnosis system for consistency checks.

A combined model is arranged in state space form to be used in the controller design. Linearizations of the nonlinear parts of the model are derived from this state space form. Additional information about the model is found in [9].

2.7. Model simplifications

The following list outlines the major simplifications of the model.

- *Rigid structure*: The rotor is assumed rigid, such that the stiff blades are fixed to the hub, which is fixed to the low-speed shaft. This simplification eliminates bending modes of the blades. Additionally, a stiff tower is assumed and sideward tower movements are neglected.
- *Fixed environmental variables*: The wind is assumed to be perpendicular to the rotor plane at all times, eliminating the impact of yaw misalignment. Furthermore, air density and wind shear are assumed constant.
- *Static aerodynamic model*: The aerodynamics is assumed to possess static properties, neglecting dynamic stall and dynamic inflow models.

3. Reference controller

The reference controller is supposed to approximate the configuration of an existing wind turbine control system, and is designed using classical control techniques to provide a reference for the LPV controllers. The controller parameters are found in Appendix B. Details regarding the design and evaluation of the reference controller are given in [9].

In full load operation two PI-controllers are used to track a constant generator speed reference and a constant power reference, see Fig. 1. A speed controller controls the pitch angle of the blades while a power controller controls the generator torque. The generator torque is kept close to a nominal value and resulting speed variations are compensated by the pitch system. Therefore, pitch control is the essential element in full load operation. A diagram of the reference control system is shown in Fig. 5. Both PI-controllers can be expressed on this general form, where s denotes the Laplace operator:

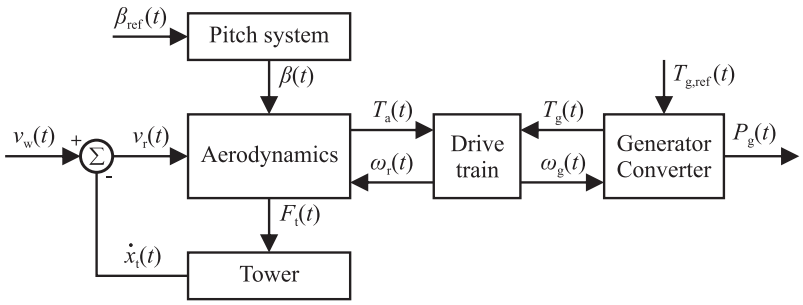

Fig. 4. Block diagram of the wind turbine model. Measurements are emulated by sampling $T_g(t)$, $P_g(t)$, $\beta(t)$, $\omega_g(t)$, and $\omega_r(t)$ at 100 Hz and then adding zero-mean white Gaussian noise.

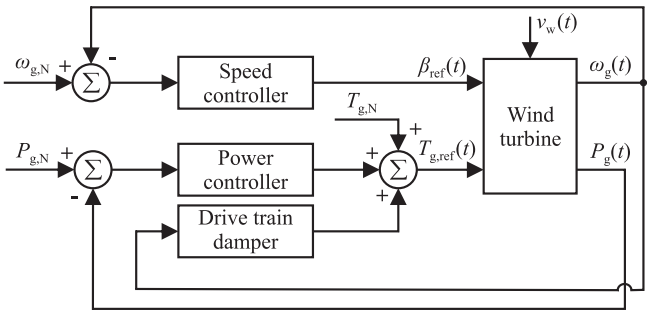

Fig. 5. Reference control system for the full load region tracking a nominal generator speed reference, $\omega_{g,N}$, and a nominal power reference, $P_{g,N}$. The mean nominal generator torque is $T_{g,N}$.

$$D(s) = K\left(1 + \frac{1}{T_i \cdot s}\right) \tag{11}$$

The gain-scheduled speed controller consists of two controllers operating at wind speeds of 12–15 m/s and 15–25 m/s, respectively. These controllers were designed to have a gain margin of 6 dB and a phase margin of 50°, with further details given in [9]. A simple bumpless transfer was introduced to schedule between the two controllers, utilizing the pitch angle as scheduling parameter.

In order to mitigate the effects of drive train oscillations, the measured generator speed and power are band-stop filtered before they are fed to the PI-controllers. The filters are introduced in order to remove the drive train eigenfrequency from the measurements. Additionally, active drive train damping is deployed by adding a signal to the generator torque to compensate for the oscillations in the drive train. This signal should have a frequency, ω_{nd}, equal to the eigenfrequency of the drive train, which is obtained by filtering the measurement of the generator speed using the filter:

$$D_d(s) = K_d \frac{2\zeta_d \omega_{nd} s(1 + \tau_d s)}{s^2 + 2\zeta_d \omega_{nd} s + \omega_{nd}^2} \quad [\mathrm{Nm/(rad/s)}] \tag{12}$$

The time constant, τ_d, introduces a zero in the filter, and can be used to compensate for time lags in the system. Therefore, τ_d is selected to equal the time delay of the converter system, $t_{g,d}$. The filter gain is K_d and the damping ratio is ζ_d. The active drive train damping and power controller were also used for the LPV controllers.

4. Controller design

This section presents the LMI-based method for designing and synthesizing the four controllers introduced in Section 1. The four controllers are: Nominal Controller (NC), Active Fault-Tolerant Controller (AFTC), Passive Fault-Tolerant Controller (PFTC), and Robust Controller (RC).

Wind turbines inherently exhibit nonlinear dynamics that are highly dependent on the current operating condition of the wind turbine. LPV control is utilized in order to schedule the four controllers according to the nominal operating trajectory of the full load region shown in Fig. 1. Additionally, one controller is designed to be robust in order to guarantee stability and performance despite the nonlinear nature of the system. Since wind turbines are also exposed to faults, two different fault-tolerant LPV controllers are furthermore designed. As mentioned above, several controllers will be designed in this paper, where each of them implements one or more of the desired features. We shall, however, apply a joint LPV controller design framework, for which all of the individual controller designs can be seen as special cases. To support this we shall extend the LPV controller design in [13] in order to allow for unmeasured parameter variations in the same framework. In contrast to [14], the method relies on a structured uncertainty description and is based on parameter dependent Lyapunov functions as in [15], and the rates of the parameter variations are bounded as in [16].

A block diagram of the wind turbine being controlled by an LPV controller is illustrated in Fig. 6, where $u(k)$ is the control signal and $w(k)$ is the disturbance. The LPV controllers depend on the measurements $y(k)$ and an estimate of the current operating point, $\hat{\theta}_{op}(k)$, which is used as scheduling parameter. Additionally, a fault diagnosis system provides the scheduling parameter $\hat{\theta}_f(k)$ for the active fault-tolerant controller. The extra degree of freedom added by allowing the AFTC to adapt in case of a fault may introduce less conservatism than for the PFTC. However, if the fault diagnosis system behaves incorrectly, the AFTC is affected in an undesirable manner. The generation of the scheduling parameters is explained in Section 4.2.

Compared to the reference controller, the LPV controller manages multiple inputs, which is generally a key to improving performance of a control system. In the considered case, the noisy rotor speed measurement only slightly improves the estimated rotational speed. Additionally, knowledge of the pitch angle measurement gives only a slight improvement of the estimated pitch angle, as the pitch angle reference is already known to the controller.

The AFTC is a conventional LPV controller scheduled on $\theta_{op}(t)$ and $\theta_f(t)$; the reason for denoting it an active fault-tolerant controller arises from the origin of the scheduling parameters. Throughout this paper $\theta(t)$ represents the measured parameter variations and $\Delta(t)$ represents the unmeasured parameter variations.

The nominal controller and the AFTC are designed using the LPV controller design method described in [13]. This method is based on output feedback, which suits the considered problem well since the state vector is only partially measured. To enable the design of the robust controller and the PFTC, the description in [13] is extended in this paper by allowing unmeasured parameter variations in the design.

4.1. System and controller description

The wind turbine model is given by the general LPV system description shown in (13), where the subscripts θ and Δ are used as shorthand notation for a matrix depending on $\theta(t)$ and $\Delta(t)$, i.e. $A(\theta, \Delta)$ is denoted $A_{\theta\Delta}$. Note that $z(t)$ is the performance output vector.

$$\dot{x}(t) = A_{\theta\Delta}x(t) + B_{1\theta\Delta}w(t) + B_{2\theta\Delta}u(t) \tag{13a}$$
$$z(t) = C_{1\theta\Delta}x(t) + D_{11\theta\Delta}w(t) + D_{12\theta\Delta}u(t) \tag{13b}$$
$$y(t) = C_{2\theta\Delta}x(t) + D_{21\theta\Delta}w(t) + D_{22\theta\Delta}u(t) \tag{13c}$$

The unmeasured parameter vector Δ is empty for the nominal controller and the active fault-tolerant controller, since all scheduling parameters are measured in these cases.

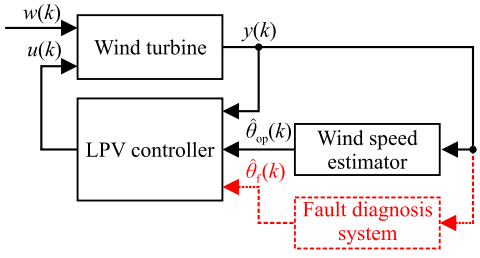

Fig. 6. Block diagram of the controller structures. The black boxes are common to the LPV controllers, while the red dashed box illustrates the fault diagnosis system required by the AFTC. The LPV controller replaces the speed controller in Fig. 5.

The controller description in (14), where the controller matrices are dependent on the measured parameters, $\theta(t)$, is common to all of the controllers.

$$\dot{x}_c(t) = A_{c\theta}x_c(t) + B_{c\theta}y(t) \tag{14a}$$
$$u(t) = C_{c\theta}x_c(t) + D_{c\theta}y(t) \tag{14b}$$

The size of the optimization problem involved in the controller design is connected to how the model depends on θ and Δ. An affine dependence is preferable, otherwise grid-based methods should be used at high computational cost; for details see Appendix B in [6].

4.2. Scheduling parameters

For each controller a system description dependent on the varying parameters was set up, which involves determining the ranges and rate bounds of the scheduling parameters. All controllers were designed to operate in the entire full load region, ranging between wind speeds of 13 m/s and 25 m/s, extending the results in [17].

The parameter variations of the system along the nominal operating trajectory originate from the linearized version of the aerodynamic torque shown in (15), which is a linearization of (1).

$$\begin{aligned} T_a(t) &\approx \overline{T}_a + T_a^{\Delta}(t) \\ &\approx \overline{T}_a + \frac{\partial T_a}{\partial \omega_r}\omega_r^{\Delta}(t) + \frac{\partial T_a}{\partial v_r}v_r^{\Delta}(t) + \frac{\partial T_a}{\partial \beta}\beta^{\Delta}(t) \quad [\text{Nm}] \end{aligned} \tag{15}$$

where $\frac{\partial T_a}{\partial \omega_r}$, $\frac{\partial T_a}{\partial v_r}$, and $\frac{\partial T_a}{\partial \beta}$ are instantaneous partial derivatives of the aerodynamic torque and ω_r^{Δ}, v_r^{Δ}, and β^{Δ} are the deviations from the operating point where T_a is linearized.

The parameter variations in the nominal LPV plant model were approximated using an affine description in the wind speed, i.e. $\theta_{op}(t) = v_r(t)$. The affine approximations of the instantaneous partial derivatives of the aerodynamic torque are displayed in Fig. 7. By inspecting the output of the wind model, the rate bounds of $v_r(t)$ were approximated to be -2 m/s^2 and 2 m/s^2.

In wind turbine control it is highly desirable to know the effective wind speed, which is defined as the spatial average of the wind field over the rotor plane with the wind stream being unaffected by the wind turbine. However, on most wind turbines the wind speed is measured by an anemometer on the nacelle, which only measures the wind speed at a single point in space and is affected by the presence of the rotor. This measurement is not representative of the effective wind speed. To obtain the scheduling parameter, an effective wind speed estimator was therefore designed according to the method in [18], described in detail in [9].

4.2.1. Fault-tolerant controllers

Since the parameter variations of the pitch system introduced in (5) are affine in θ_f and can be used directly as shown in (6), the fault in the pitch system can be incorporated into the LPV description forthwith. The fault signal ranges between zero and one for convenience and may be fully introduced within 30 s, corresponding to a rapid drop in pressure of the pitch system. This makes the parameter range and the rate bounds be $\theta_f \in [0,1]$ and $\dot{\theta}_f \in [-0.033/\text{s},\ 0.033/\text{s}]$.

The scheduling parameter θ_f was not measured and had to be estimated for the AFTC. Therefore, a parameter estimator was designed based on a multi-model estimation method using an extended Kalman filter explained in [19], and relies on the measured pitch angles of all three blades. The design of the fault diagnosis system is described in detail in [9].

4.2.2. Robust Controller

The affine approximations of the partial derivatives of the aerodynamic torque along the nominal operating trajectory are shown in Fig. 7. Although the approximations are reasonably good, the deviations might still require some robustness of the controller. Additionally, it is not expected that the nominal operating trajectory is followed exactly even at normal operation due to the stochastic wind input and measurement noise. Combined with the simplification of a static aerodynamic model, robustness towards parameter uncertainties in the aerodynamic model is of great importance to the controller.

The uncertainty in the partial derivatives is modeled by adding three independent uncertain but bounded variables Δ_1, Δ_2, and Δ_3 to the parameter description:

$$\frac{\partial T_a}{\partial \beta}(v_r, \Delta_1) \approx a_\beta + b_\beta v_r + \Delta_1 \tag{16a}$$
$$\frac{\partial T_a}{\partial v_r}(v_r, \Delta_2) \approx a_{v_r} + b_{v_r} v_r + \Delta_2 \tag{16b}$$
$$\frac{\partial T_a}{\partial \omega_r}(v_r, \Delta_3) \approx a_{\omega_r} + b_{\omega_r} v_r + \Delta_3 \tag{16c}$$

To determine the bounds on Δ_1, Δ_2, and Δ_3 the reference controller was simulated with wind speeds ranging from 13 m/s to 25 m/s to record the actual partial derivatives of the aerodynamic torque. These are illustrated in Fig. 8 by the green points. It was decided for the design of the robust controller to use the parameter set marked by the red boxes, which covers the majority of the samples; hence, the parameter ranges are $\Delta_1 \in [-0.0994, 0.0426]$ MNm/°, $\Delta_2 \in [-0.104, 0.104]$ MNm/(m/s), and $\Delta_3 \in [-1.0, 1.1]$ MNm/(rad/s). The rates of the parameter variations of Δ_1, Δ_2, and Δ_3 were not determined, since they are not utilized in the controller design.

4.2.3. Affine system description

From the previous description of the parameter dependencies it is concluded that an affine parameter description can be adapted for the considered system by simplifying the general case in [13], where no restrictions are imposed on the parameter dependence.

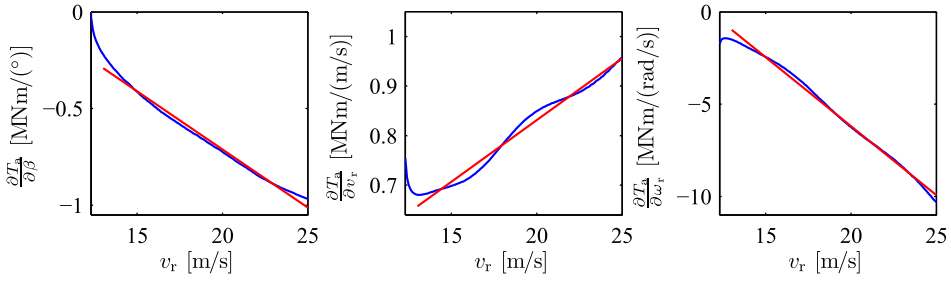

Fig. 7. Parameter variations along the nominal operating trajectory in the full load region. The partial derivatives of the aerodynamic torque (blue) are approximated in the range from 13 m/s to 25 m/s by affine descriptions (red) using the wind speed as scheduling parameter. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

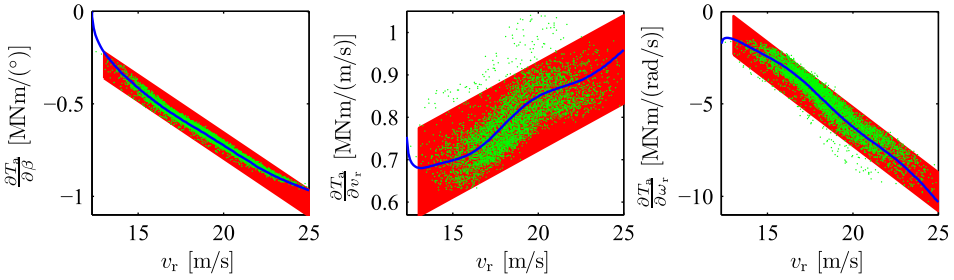

Fig. 8. Parameter variations along the nominal operating trajectory in the full load region. Partial derivatives of the aerodynamic torque in simulation of the reference controller (green) are used in the design of the robust controller. The partial derivatives of the aerodynamic torque (blue) are approximated in the range from 13 m/s to 25 m/s by affine uncertain descriptions (red) using the wind speed as scheduling parameter and three uncertain parameters. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The affine description reduces the number of matrix inequalities in the optimization problem used to compute the controller matrices. The system description is shown in (17), where n_θ is the number of measured parameters and n_Δ is the number of unmeasured parameters. To ease the design of the LPV controller, some matrices in the LPV system description are assumed to be constant. However, the restriction of parameter independent matrices B_2, C_2, D_{12}, and D_{21} can be applied without any loss of generality, see [20]. Furthermore, the assumption $D_{22} = 0$ can be relaxed by redefining the output.

$$\begin{bmatrix} A_{\theta\Delta} & B_{1\theta\Delta} & B_{2\theta\Delta} \\ C_{1\theta\Delta} & D_{11\theta\Delta} & D_{12\theta\Delta} \\ C_{2\theta\Delta} & D_{21\theta\Delta} & D_{22\theta\Delta} \end{bmatrix} = \begin{bmatrix} A_0 & B_{1,0} & B_{2,0} \\ C_{1,0} & D_{11,0} & D_{12,0} \\ C_{2,0} & D_{21,0} & 0 \end{bmatrix} + \sum_{i=1}^{n_\theta} \theta_i \begin{bmatrix} A_i^\theta & B_{1,i}^\theta & 0 \\ C_{1,i}^\theta & D_{11,i}^\theta & 0 \\ 0 & 0 & 0 \end{bmatrix} + \sum_{j=1}^{n_\Delta} \Delta_j \begin{bmatrix} A_j^\Delta & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{17}$$

If an affine approximation cannot be justified, uncertain parameters can be introduced to cover model nonlinearities, as in the section presenting robustness against model uncertainties. This however introduces conservatism, in particular for highly nonlinear systems.

For convenience the matrices A_θ and A_Δ^{lin} are defined in (18), describing the measured and unmeasured parameter variations.

$$A_{\theta\Delta} = \underbrace{A_0 + \sum_{i=1}^{n_\theta} \theta_i A_i^\theta}_{A_\theta} + \underbrace{\sum_{j=1}^{n_\Delta} \Delta_j A_j^\Delta}_{A_\Delta^{\text{lin}}} \tag{18}$$

The NC and AFTC have no unmeasured parameters. Correspondingly, for the cases under consideration:

$$\text{NC} : A_{\theta\Delta} = A_0 + \underbrace{\theta_1}_{v_r} A_1^\theta \tag{19a}$$

$$\text{AFTC} : A_{\theta\Delta} = A_0 + \underbrace{\theta_1}_{v_r} A_1^\theta + \underbrace{\theta_2}_{\theta_f} A_2^\theta \tag{19b}$$

$$\text{PFTC} : A_{\theta\Delta} = A_0 + \underbrace{\theta_1}_{v_r} A_1^\theta + \underbrace{\Delta_4}_{\theta_f} A_4^\Delta \tag{19c}$$

$$\text{RC} : A_{\theta\Delta} = A_0 + \underbrace{\theta_1}_{v_r} A_1^\theta + \Delta_1 A_1^\Delta + \Delta_2 A_2^\Delta + \Delta_3 A_3^\Delta \tag{19d}$$

Note that θ_2 and Δ_4 describe the exact same parameter variation of the system. In the AFTC case, the parameter θ_2 is considered to be measured, whereas in the PFTC case, the parameter Δ_4 is considered to be an uncertainty, which the controller must provide robustness for.

The general LPV plant model is shown in (20) and for the four controller designs it is adapted to the cases in (19). To ease the controller design by fulfilling (17), B_2 is turned into a parameter independent matrix by performing a state transformation which replaces the state $\dot\beta(t)$ with $\dot\beta'(t) = \frac{1}{\omega_n^2(t)}\dot\beta(t)$.

$$\dot x(t) = A(\theta)x(t) + B_1(\theta)w(t) + B_2 u(t)$$

$$\begin{bmatrix} \dot\beta(t) \\ \ddot\beta'(t) \\ \dot\theta_\Delta(t) \\ \dot\omega_g(t) \\ \dot\omega_r(t) \end{bmatrix} = \begin{bmatrix} 0 & a_{12}(\theta_f) & 0 & 0 & 0 \\ -1 & a_{22}(\theta_f) & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{N_g} & 1 \\ 0 & 0 & \frac{K_{dt}}{J_g N_g} & -\left(\frac{B_{dt}}{J_g N_g^2} + \frac{B_g}{J_g}\right) & \frac{B_{dt}}{N_g J_g} \\ \frac{1}{J_r}\frac{\partial T_a(\theta_{op})}{\partial \beta} & 0 & -\frac{K_{dt}}{J_r} & \frac{B_{dt}}{N_g J_r} & a_{55}(\theta_{op}) \end{bmatrix} \begin{bmatrix} \beta(t) \\ \dot\beta'(t) \\ \theta_\Delta(t) \\ \omega_g(t) \\ \omega_r(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{J_r}\frac{\partial T_a(\theta_{op})}{\partial v_r} \end{bmatrix} v_r(t) + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \beta_{\text{ref}}(t - t_d) \tag{20}$$

$$a_{12}(\theta_f) = (1 - \theta_f(t))\omega_{n,0}^2 + \theta_f(t)\omega_{n,lp}^2$$

$$a_{22}(\theta_f) = -2(1 - \theta_f(t))\zeta_0\omega_{n,0} - 2\theta_f(t)\zeta_{lp}\omega_{n,lp}$$

$$a_{55}(\theta_{op}) = \frac{1}{J_r}\frac{\partial T_a(\theta_{op})}{\partial \omega_r} - \frac{B_{dt} + B_r}{J_r}$$

The LPV controllers rely on measurements of $\beta(t)$, $\omega_r(t)$, and $\omega_g(t)$. Additionally, the AFTC also receives the fault estimate $\hat\theta_f(t)$ from the fault diagnosis system.

4.3. LPV controller design method

From the system description in (20) and the LPV controller in (14), the design task is to find a parameter-dependent quadratic (PDQ) stable closed-loop system, which minimizes the induced $\mathcal{L}_2$-norm between the disturbance input, $w(t)$, and the performance output, $z(t)$. This can be accomplished by finding parameter-dependent Lyapunov functions, as explained in the following theorem presented in [6] originating from [13]. Note that the subscript 'cl' denotes closed-loop system matrices.

Theorem 1. *Given a closed-loop system governed by the parameter-dependent matrices $A_{cl\theta\Delta}$, $B_{cl\theta}$, $C_{cl\theta}$, $D_{cl\theta}$ with $(\theta, \dot\theta, \Delta, \dot\Delta) \in \Theta \times \mathcal{V} \times \mathcal{D} \times \mathcal{U}$, suppose that there exists a differentiable symmetric function $X_{cl\theta}$ such that $X_{cl\theta} > 0$ and*

$$\begin{bmatrix} \dot X_{cl\theta} + A_{cl\theta\Delta}^T X_{cl\theta} + X_{cl\theta} A_{cl\theta\Delta} & X_{cl\theta} B_{cl\theta} & C_{cl\theta}^T \\ B_{cl\theta}^T X_{cl\theta} & -\gamma I & D_{cl\theta}^T \\ C_{cl\theta} & D_{cl\theta} & -\gamma I \end{bmatrix} < 0 \tag{21}$$

for all $(\theta, \dot{\theta}, \Delta, \dot{\Delta}) \in \Theta \times \mathcal{V} \times \mathcal{D} \times \mathcal{U}$. *Then,*

1. the function $A_{\mathrm{cl}\theta\Delta}$ is PDQ stable over $\Theta \times \mathcal{D}$, and
2. the induced $\mathcal{L}_2$-norm of the operator T_{zw} is bounded by $\gamma > 0$.

A bound γ on $\|T_{zw}\|_{i,2}$ equivalent to $\|T_{zw}\|_{i,2} < \gamma$ means that

$$\int_0^\infty z^{\mathrm{T}}(\tau) z(\tau) d\tau < \gamma^2 \int_0^\infty w^{\mathrm{T}}(\tau) w(\tau) d\tau \tag{22}$$

Theorem 1 cannot be utilized directly for controller design, since the closed-loop system matrices are unknown. To form an appropriate design problem, which can be solved using convex optimization, some auxiliary controller matrices are defined as shown in (23). Notice that the bold symbols are unknown matrices in the design problem.

$$\begin{aligned}\widehat{\boldsymbol{A}}_\theta &= N_\theta A_{\mathrm{c}\theta} M_\theta^{-\mathrm{T}} - \boldsymbol{X}_\theta \dot{\boldsymbol{Y}}_\theta - N_\theta \dot{M}_\theta^{\mathrm{T}} + \boldsymbol{X}_\theta B_2 C_{\mathrm{c}\theta} M_\theta^{\mathrm{T}} + N_\theta B_{\mathrm{c}\theta} C_2 \boldsymbol{Y}_\theta \\ &\quad + \boldsymbol{X}_\theta (A_\theta + B_2 D_{\mathrm{c}\theta} C_2) \boldsymbol{Y}_\theta \end{aligned} \tag{23a}$$

$$\widehat{\boldsymbol{B}}_\theta = N_\theta B_{\mathrm{c}\theta} + \boldsymbol{X}_\theta B_2 D_{\mathrm{c}\theta} \tag{23b}$$

$$\widehat{\boldsymbol{C}}_\theta = C_{\mathrm{c}\theta} M_\theta^{\mathrm{T}} + D_{\mathrm{c}\theta} C_2 \boldsymbol{Y}_\theta \tag{23c}$$

$$\widehat{\boldsymbol{D}}_\theta = D_{\mathrm{c}\theta} \tag{23d}$$

Due to the assumption of an affine parameter description, the Lyapunov matrices $\boldsymbol{X}_\theta$ and $\boldsymbol{Y}_\theta$, and auxiliary matrices in (23) $\widehat{\boldsymbol{A}}_\theta$, $\widehat{\boldsymbol{B}}_\theta$, $\widehat{\boldsymbol{C}}_\theta$, $\widehat{\boldsymbol{D}}_\theta$, are described using an affine description:

$$\boldsymbol{X}_\theta = \boldsymbol{X}_0 + \sum_{i=1}^{n_\theta} \theta_i \boldsymbol{X}_i \;\; \boldsymbol{Y}_\theta = \boldsymbol{Y}_0 + \sum_{i=1}^{n_\theta} \theta_i \boldsymbol{Y}_i \;\; \widehat{\boldsymbol{A}}_\theta = \widehat{\boldsymbol{A}}_0 + \sum_{i=1}^{n_\theta} \theta_i \widehat{\boldsymbol{A}}_i \tag{24a}$$

$$\widehat{\boldsymbol{B}}_\theta = \widehat{\boldsymbol{B}}_0 + \sum_{i=1}^{n_\theta} \theta_i \widehat{\boldsymbol{B}}_i \;\; \widehat{\boldsymbol{C}}_\theta = \widehat{\boldsymbol{C}}_0 + \sum_{i=1}^{n_\theta} \theta_i \widehat{\boldsymbol{C}}_i \;\; \widehat{\boldsymbol{D}}_\theta = \widehat{\boldsymbol{D}}_0 + \sum_{i=1}^{n_\theta} \theta_i \widehat{\boldsymbol{D}}_i \tag{24b}$$

From (21), (23), and (24), Theorem 1 is reformulated into:

Theorem 2. *Given the open-loop LPV system in (13) with matrices defined in (17), suppose that there exists two parameter-dependent symmetric matrices* X_θ *and* Y_θ *and four parameter-dependent matrices* $\widehat{\boldsymbol{A}}_\theta$, $\widehat{\boldsymbol{B}}_\theta$, $\widehat{\boldsymbol{C}}_\theta$, $\widehat{\boldsymbol{D}}_\theta$, *defined in (24), such that for all* $(\theta, \dot{\theta}, \Delta, \dot{\Delta}) \in \Theta \times \mathcal{V} \times \mathcal{D} \times \mathcal{U}$,

$$\begin{bmatrix} \phi_{11} & * & * & * \\ \phi_{21} & \phi_{22} & * & * \\ (\boldsymbol{X}_\theta B_{1\theta} + \widehat{\boldsymbol{B}}_\theta D_{21})^{\mathrm{T}} & (B_{1\theta} + B_2 \widehat{\boldsymbol{D}}_\theta D_{21})^{\mathrm{T}} & -\gamma I_{n_w} & * \\ C_{1\theta} + D_{12} \widehat{\boldsymbol{D}}_\theta C_2 & C_{1\theta} \boldsymbol{Y}_\theta + D_{12} \widehat{\boldsymbol{C}}_\theta & D_{11\theta} + D_{12} \widehat{\boldsymbol{D}}_\theta D_{21} & -\gamma I_{n_z} \end{bmatrix} < 0 \tag{25a}$$

$$\begin{aligned} \phi_{11} &= \dot{\boldsymbol{X}}_\theta + \boldsymbol{X}_\theta A_{\theta\Delta} + \widehat{\boldsymbol{B}}_\theta C_2 + A_{\theta\Delta}^{\mathrm{T}} \boldsymbol{X}_\theta + C_2^{\mathrm{T}} \widehat{\boldsymbol{B}}_\theta^{\mathrm{T}} \\ \phi_{21} &= \widehat{\boldsymbol{A}}_\theta^{\mathrm{T}} + \boldsymbol{Y}_\theta \left(A_\Delta^{\mathrm{lin}}\right)^{\mathrm{T}} \boldsymbol{X}_\theta + A_{\theta\Delta} + B_2 \widehat{\boldsymbol{D}}_\theta C_2 \\ \phi_{22} &= -\dot{\boldsymbol{Y}}_\theta + A_{\theta\Delta} \boldsymbol{Y}_\theta + B_2 \widehat{\boldsymbol{C}}_\theta + \boldsymbol{Y}_\theta A_{\theta\Delta}^{\mathrm{T}} + \widehat{\boldsymbol{C}}_\theta^{\mathrm{T}} B_2^{\mathrm{T}} \\ \begin{bmatrix} \boldsymbol{X}_\theta & I \\ I & \boldsymbol{Y}_\theta \end{bmatrix} &> 0 \end{aligned} \tag{25b}$$

Then, there exists a controller of the form in (14) such that

1. *the closed-loop system is PDQ stable over* $\Theta \times \mathcal{D}$ *and,*
2. *the induced* $\mathcal{L}_2$*-norm of the operator* T_{zw} *is bounded by* $\gamma > 0$.

In large symmetric matrix expressions, terms denoted '$*$' will be induced by symmetry. For instance, with S and P symmetric

$$\begin{bmatrix} S + M + N + (*) & * \\ Q & P \end{bmatrix} = \begin{bmatrix} S + M + M^{\mathrm{T}} + N + N^{\mathrm{T}} & Q^{\mathrm{T}} \\ Q & P \end{bmatrix}$$

From Theorem 2 it is seen that the inequalities should hold in the entire parameter space. However, we want to solve a finite set of inequalities. It is only necessary to test the matrix inequalities (25a) and (25b) in the vertices of the parameter space, Δ_{vex}, if the following additional constraint is satisfied:

$$\begin{bmatrix} \boldsymbol{X}_i A_i^\theta + \left(A_i^\theta\right)^{\mathrm{T}} \boldsymbol{X}_i & * & * & * \\ \boldsymbol{Y}_i \left(A_\Delta^{\mathrm{lin}}\right)^{\mathrm{T}} \boldsymbol{X}_i & A_i^\theta \boldsymbol{Y}_i + \boldsymbol{Y}_i \left(A_i^\theta\right)^{\mathrm{T}} & * & * \\ \left(B_{1,i}^\theta\right)^{\mathrm{T}} \boldsymbol{X}_i & 0 & 0 & * \\ 0 & C_{1,i}^\theta \boldsymbol{Y}_i & 0 & 0 \end{bmatrix} \geqslant 0 \tag{25c}$$

for $i = 1, \ldots, n_\theta$ and $\Delta \in \Delta_{\mathrm{vex}}$.

It appears from the structure of (25c) that $\boldsymbol{X}_i$ should be in the null space of $\left(B_{1,i}^\theta\right)^{\mathrm{T}}$ and $\boldsymbol{Y}_i$ should be in the null space of $C_{1,i}^\theta$, otherwise the matrix is indefinite. The additional LMI introduces some conservatism, which is dependent on how sparse the matrices $B_{1,i}^\theta$ and $C_{1,i}^\theta$ are. If e.g. $B_{1,i}^\theta$ is a zero matrix there are no additional constraints on $\boldsymbol{X}_i$. Usually, either $\boldsymbol{X}_\theta$ or $\boldsymbol{Y}_\theta$ is selected to be independent of $\theta(t)$, i.e. either $\boldsymbol{X}_i$ or $\boldsymbol{Y}_i$ is a zero matrix. The reason for this is that if both $\boldsymbol{X}_\theta$ and $\boldsymbol{Y}_\theta$ depend on θ, then $\dot{\theta}(t)$ should be measured to synthesize the controller [13].

For NC and AFTC A_Δ^{lin} is a zero matrix, turning the optimization problem into an LMI-based optimization problem, since the term $\boldsymbol{Y}_\theta \left(A_\Delta^{\mathrm{lin}}\right)^{\mathrm{T}} \boldsymbol{X}_\theta$ vanishes in (25). In this case, the optimization problem is convex and the controller giving the smallest γ can be found easily using convex optimization. In contrast, the optimization problems for PFTC and RC are based on bilinear matrix inequalities (BMIs) due to non-zero elements in A_Δ^{lin}; hence, some additional work must be done to solve these problems.

4.4. Solving the BMI-based optimization problems

To solve the BMI-based optimization problem for the passive fault-tolerant controller and the robust controller, a two-step procedure is suggested inspired by Jabbari [21], where the projection lemma, provided hereafter, is utilized to derive two necessary LMI conditions (27) and (28), for the BMIs in (25).

Lemma 1 (Projection lemma). *Given a symmetric matrix* Ω *and matrices* $\mathcal{B}$ *and* $\mathcal{C}$ *of compatible dimensions, there exists a matrix* $\mathcal{L}$ *such that* $\Omega + \mathcal{B}\mathcal{L}\mathcal{C} + (\mathcal{B}\mathcal{L}\mathcal{C})^{\mathrm{T}} < 0$ *if and only if*

$$\mathcal{B}_\perp^{\mathrm{T}} \Omega \mathcal{B}_\perp < 0 \quad \text{and} \tag{26a}$$

$$(\mathcal{C}^{\mathrm{T}})_\perp^{\mathrm{T}} \Omega (\mathcal{C}^{\mathrm{T}})_\perp < 0, \tag{26b}$$

where $\mathcal{B}_\perp$ *is defined as a basis for the null space of* $\mathcal{B}^{\mathrm{T}}$.

Lemma 1 is utilized to solve (25), corresponding to $\Omega + \mathcal{B}\mathcal{L}\mathcal{C} + (\mathcal{B}\mathcal{L}\mathcal{C})^{\mathrm{T}} < 0$, by exploiting the necessary LMI conditions derived in (27)–(28), corresponding to (26). Since the necessary conditions are LMIs they can be solved using conventional convex methods. The following algorithm is utilized to solve (25):

1. Solve one of the necessary conditions (27) or (28) defined below. Solving a necessary condition provides some of the unknown variables of (25), in the considered case making it an LMI in the unknown variables.
2. Solve (25), where the variables from the previous step are utilized.

The necessary conditions for (25) are set up below.

Necessary condition for $\boldsymbol{X}_\theta$*:*

$$\begin{bmatrix} \phi_{11} & * & * \\ (\boldsymbol{X}_\theta B_{1\theta} + \widehat{\boldsymbol{B}}_\theta D_{21})^{\mathrm{T}} & -\gamma I_{n_w} & * \\ C_{1\theta} + D_{12} \widehat{\boldsymbol{D}}_\theta C_2 & D_{11\theta} + D_{12} \widehat{\boldsymbol{D}}_\theta D_{21} & -\gamma I_{n_z} \end{bmatrix} < 0 \tag{27a}$$

for all $\theta \in \theta_{\text{vex}}, \Delta \in \Delta_{\text{vex}}, \dot{\theta} \in \dot{\theta}_{\text{vex}}$, *and*

$$\begin{bmatrix} \boldsymbol{X}_i A_i^\theta + \left(A_i^\theta\right)^{\mathrm{T}} \boldsymbol{X}_i & * & * \\ \left(B_{1,i}^\theta\right)^{\mathrm{T}} \boldsymbol{X}_i & 0 & * \\ 0 & 0 & 0 \end{bmatrix} \geqslant 0 \tag{27b}$$

for $i = 1, \ldots, n_\theta$.

Necessary condition for $\boldsymbol{Y}_\theta$:

$$\begin{bmatrix} \phi_{22} & * & * \\ (B_{1\theta} + B_2 \widehat{\boldsymbol{D}}_\theta D_{21})^{\mathrm{T}} & -\gamma I_{n_w} & * \\ C_{1\theta} \boldsymbol{Y}_\theta + D_{12} \widehat{\boldsymbol{C}}_\theta & D_{11\theta} + D_{12} \widehat{\boldsymbol{D}}_\theta D_{21} & -\gamma I_{n_z} \end{bmatrix} < 0 \tag{28a}$$

for all $\theta \in \theta_{\text{vex}}, \Delta \in \Delta_{\text{vex}}, \dot{\theta} \in \dot{\theta}_{\text{vex}}$, *and*

$$\begin{bmatrix} A_i^\theta \boldsymbol{Y}_i + \boldsymbol{Y}_i \left(A_i^\theta\right)^{\mathrm{T}} & * & * \\ 0 & 0 & * \\ C_{1,i}^\theta \boldsymbol{Y}_i & 0 & 0 \end{bmatrix} \geqslant 0 \tag{28b}$$

for $i = 1, \ldots, n_\theta$.

The robust controller is designed to guarantee the same performance in the entire uncertain parameter space, whereas the PFTC is a reliable controller designed to guarantee higher performance in the normal case than in the fault case. In the latter case, this is achieved by using different γ values for the normal and faulty systems. The compromise between good performance in the normal case and good performance in the faulty case is a design choice, which is illustrated in Fig. 9, showing the Pareto optimum tradeoff curve between conflicting requirements.

The Pareto optimality tradeoff curve shows the compromise between a good performance in the normal case (small γ_{n}) verses good performance in the faulty case (small γ_{f}). If $\gamma_{\text{n}} = \gamma_{\text{f}}$ we get the same compromise as for a robust controller, which has the same performance for all situations. For the passive fault-tolerant controller it is desirable to have the best performance in the normal case, as the system is expected to be in this state most of the time. Between the two vertices, the guaranteed performance follows graceful degradation according to $\gamma = \frac{\gamma_{\text{n}}\gamma_{\text{f}}}{\gamma_{\text{n}}(1-\theta_{\text{f}})+\gamma_{\text{f}}\theta_{\text{f}}}$, where θ_{f} indicates the state of the system between 0 (normal, γ_{n}) and 1 (faulty, γ_{f}).

4.5. Controller synthesis

When the optimization problem is solved, the following synthesis procedure is used to calculate the controller matrices at each sample time:

1. Compute $\widehat{\boldsymbol{A}}_\theta$, $\widehat{\boldsymbol{B}}_\theta$, $\widehat{\boldsymbol{C}}_\theta$, $\widehat{\boldsymbol{D}}_\theta$, $\boldsymbol{X}_\theta$, and $\boldsymbol{Y}_\theta$ using the measured value of $\theta(t)$.

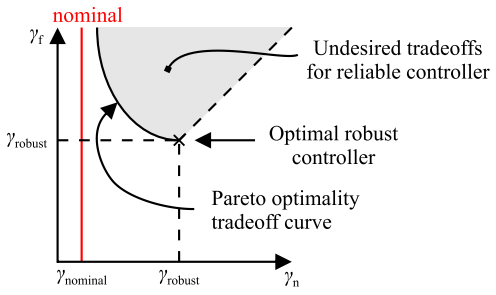

Fig. 9. Bounds on γ values for the PFTC. γ_{n} is associated with the fault-free case and γ_{f} is associated with the faulty case. The gray area contains all feasible PFTCs, where a PFTC on the Pareto optimality tradeoff curve is desired.

2. Find M_θ and N_θ by solving the factorization problem:

$$I - \boldsymbol{X}_\theta \boldsymbol{Y}_\theta = N_\theta M_\theta^{\mathrm{T}} \tag{29}$$

3. Compute $A_{\text{c}\theta\dot{\theta}}$, $B_{\text{c}\theta}$, $C_{\text{c}\theta}$, and $D_{\text{c}\theta}$ from the equations:

$$A_{\text{c}\theta\dot{\theta}} = N_\theta^{-1}\Big(\boldsymbol{X}_\theta \dot{\boldsymbol{Y}}_\theta + N_\theta \dot{M}_\theta^{\mathrm{T}} + \widehat{\boldsymbol{A}}_\theta - \widehat{\boldsymbol{B}}_\theta C_2 \boldsymbol{Y}_\theta - \boldsymbol{X}_\theta\Big(A_\theta - B_2 \widehat{\boldsymbol{D}}_\theta C_2\Big)\boldsymbol{Y}_\theta - \boldsymbol{X}_\theta B_2 \widehat{\boldsymbol{C}}_\theta\Big) M_\theta^{-\mathrm{T}} \tag{30a}$$

$$B_{\text{c}\theta} = N_\theta^{-1}\Big(\widehat{\boldsymbol{B}}_\theta - \boldsymbol{X}_\theta B_2 \widehat{\boldsymbol{D}}_\theta\Big) \tag{30b}$$

$$C_{\text{c}\theta} = \Big(\widehat{\boldsymbol{C}}_\theta - \widehat{\boldsymbol{D}}_\theta C_2 \boldsymbol{Y}_\theta\Big) M_\theta^{-\mathrm{T}} \tag{30c}$$

$$D_{\text{c}\theta} = \widehat{\boldsymbol{D}}_\theta \tag{30d}$$

According to [13] either $\boldsymbol{X}_\theta$ or $\boldsymbol{Y}_\theta$ must be held constant if the controller should be synthesized without measuring $\dot{\theta}(t)$. Furthermore, if N_θ and M_θ are chosen according to Table I in [13], dependencies of $\dot{\theta}(t)$ can be removed from the calculation of $A_{\text{c}\theta\dot{\theta}}$, replacing (30a) with (31).

$$A_{\text{c}\theta} = N_\theta^{-1}(\widehat{\boldsymbol{A}}_\theta - \widehat{\boldsymbol{B}}_\theta C_2 \boldsymbol{Y}_\theta - \boldsymbol{X}_\theta(A_\theta - B_2 \widehat{\boldsymbol{D}}_\theta C_2)\boldsymbol{Y}_\theta - \boldsymbol{X}_\theta B_2 \widehat{\boldsymbol{C}}_\theta) M_\theta^{-\mathrm{T}} \tag{31}$$

The design procedure applies for continuous systems, whereas the controller is a discrete component part of a sampled system. It was chosen to design the controller in continuous time, as the sampling frequency (100 Hz) is much higher than the highest frequency of the system (approximately 5 Hz). To ensure that the result of the design procedure can be implemented as a discretized controller, the location of the closed-loop poles in the complex s-plane were restricted using $\mathcal{D}$-stability [22]. This is done by limiting the real parts of the eigenvalues of the closed-loop system, such that $-\alpha < \text{Re}(s)$, by including the following LMI in the optimization problem:

$$A_{\text{cl}\theta\Delta}^{\mathrm{T}} \boldsymbol{X}_{\text{cl}\theta} + \boldsymbol{X}_{\text{cl}\theta} A_{\text{cl}\theta\Delta} + 2\alpha \boldsymbol{X}_{\text{cl}\theta} > 0, \quad \boldsymbol{X}_{\text{cl}\theta} > 0 \tag{32}$$

The eigenvalues are limited by setting $\alpha = 2\pi \cdot 25$ to ensure that the real parts of the eigenvalues are below 25 Hz.

This finalizes the procedure for synthesizing the controllers. The last part of this section applies the design method to the cases under consideration.

4.6. Computation of controllers

Each controller is designed by solving an optimization problem. This optimization problem is based on a system description, which is affine in the scheduling parameters, and which was derived in Section 4.2. In order to formulate the optimization problems, first a performance specification was composed.

4.6.1. Performance specification

The performance specification was based on a mixed sensitivity description, where it was chosen to specify sensitivity and control sensitivity. The mixed sensitivity description was implemented as shown in Fig. 10, where $W_{\text{S}}(s)$ is the sensitivity filter and $W_{\text{M}}(s)$ is the control sensitivity filter. In addition to the sensitivity filters, the input disturbance filter $W_{\text{D}}(s)$ band limits the exogenous input in the design and $W_{\text{N}}(s)$ adds measurement noise to the system outputs.

$W_{\text{S}}(s)$ stresses the importance of the low-frequency components of the generator speed error, $\omega_{\text{g,e}}(t)$. It has a pole at the origin to ensure integral action in the controllers to eliminate steady-state errors on the tracking of the generator speed reference. $W_{\text{M}}(s)$ is a high-pass filter weighting the control effort with the aim of penalizing fast pitch angle variations. The filter $W_{\text{D}}(s)$ is a low-pass filter capturing the dominating frequency content of the wind field, i.e. the effective wind speed. The frequency content of the wind

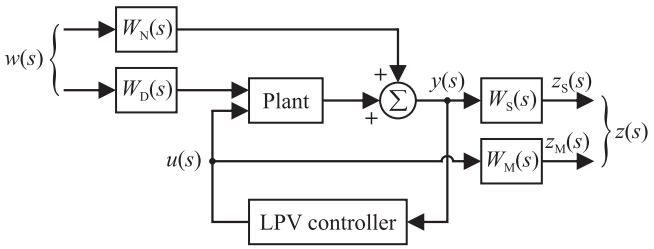

Fig. 10. Block diagram of the mixed sensitivity description.

field is related to the frequency with which the blades pass the tower. This frequency is denoted the 3P frequency, ω_{3P}, and equals three times the rotor speed, which is rated at 1.71 rad/s. Therefore, $W_D(s)$ is a low-pass filter with a pass-band up to the 3P frequency. Finally, the filter $W_N(s)$ adds noise to the measurements according to the variances listed in Section 2.6. The weighted performance measures appear in (33) and the filters are specified in (34).

$$z(s) = \begin{bmatrix} W_S(s) & 0 \\ 0 & W_M(s) \end{bmatrix} \begin{bmatrix} \omega_{g,e}(s) \\ \beta_{ref}(s) \end{bmatrix} \tag{33}$$

$$W_S(s) = k_S \frac{1}{s} \tag{34a}$$

$$W_M(s) = k_M \frac{s}{s/(10\omega_{3P}) + 1} \tag{34b}$$

$$W_D(s) = \frac{1}{s/(1.5\omega_{3P}) + 1} \tag{34c}$$

The gains of the filters were chosen to $k_M = 0.2251$ and $k_S = 0.8167$ to obtain the desired compromise between speed reference tracking and pitch angle variation. It was decided to select the gains such that the LPV controllers in the normal scenario have similar pitch actuator usage as the reference controller. This simplified the comparison of the LPV controllers and the reference controller in terms of the two performance measures. In the implementation of the sensitivity filter (34a), the system was augmented to include the integrator. For the implemented controller, the integral of the speed tracking error is measured, as in [6, p. 116].

4.6.2. Solving the optimization problems

The final task of the controller design is to solve the optimization problems for the four controllers. The optimization problem in (25) reduces to solving LMIs for NC and AFTC, as A_Δ^{lin} is a zero matrix due to the known parameter variations. The controller design problems for RC and PFTC are however BMI-based, since unknown parameter variations exist for these problems. Therefore, the following algorithm was utilized to solve the BMI-based optimization problems.

1. Select a parameter-dependent Lyapunov matrix based on the options:
   (a) Lyapunov matrix $\boldsymbol{X}$ is dependent on θ.
   (b) Lyapunov matrix $\boldsymbol{Y}$ is dependent on θ.
   (c) No Lyapunov matrix is dependent on θ.
2. Find unknown variables in (25) by solving necessary conditions, turning the matrix inequalities of (25) into LMIs in the remaining unknown variables.
   (a) Find $\boldsymbol{X}$, $\widehat{\boldsymbol{B}}_\theta$, $\widehat{\boldsymbol{D}}_\theta$ by solving the necessary condition for $\boldsymbol{X}$ shown in (27).
   (b) Find $\boldsymbol{Y}$, $\widehat{\boldsymbol{C}}_\theta$, $\widehat{\boldsymbol{D}}_\theta$ by solving the necessary condition for $\boldsymbol{Y}$ shown in (28).
3. Solve (25) while minimizing $\gamma_n = \gamma_f$ until the desired value of γ_f is reached. This is done by switching between having $\boldsymbol{X}$, $\widehat{\boldsymbol{A}}_\theta$, $\widehat{\boldsymbol{B}}_\theta$, $\widehat{\boldsymbol{C}}_\theta$, $\widehat{\boldsymbol{D}}_\theta$ or $\boldsymbol{Y}$, $\widehat{\boldsymbol{A}}_\theta$, $\widehat{\boldsymbol{B}}_\theta$, $\widehat{\boldsymbol{C}}_\theta$, $\widehat{\boldsymbol{D}}_\theta$ as unknown variables.

Table 1
γ values for the four designed controllers. The columns γ_n and γ_f indicate the performance to be expected in the normal and faulty cases.

Controller	γ_n	γ_f
Nominal controller (NC)	1.00	–
Robust controller (RC)	1.97	–
Active fault-tolerant controller (AFTC)	1.76	1.76
Passive fault-tolerant controller (PFTC)	2.30	6.45

4. Solve (25) while minimizing γ_n and switching between having $\boldsymbol{X}$, $\widehat{\boldsymbol{A}}_\theta$, $\widehat{\boldsymbol{B}}_\theta$, $\widehat{\boldsymbol{C}}_\theta$, $\widehat{\boldsymbol{D}}_\theta$ or $\boldsymbol{Y}$, $\widehat{\boldsymbol{A}}_\theta$, $\widehat{\boldsymbol{B}}_\theta$, $\widehat{\boldsymbol{C}}_\theta$, $\widehat{\boldsymbol{D}}_\theta$ as unknown variables, until γ_n stops decreasing.

From the algorithm it is seen that first it should be decided which of the Lyapunov matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ should dependent on θ. After this has been determined, one of the two necessary conditions should be utilized to find variables to turn (25) into LMIs. Note that this initialization of the algorithm may influence the final outcome of the optimization problem. The final step is to solve (25) having two different sets of unknown variables. It is chosen to solve (25) multiple times, since there is no guarantee that even a sub-optimal γ_n is obtained in one step.

The optimization problems were set up in YALMIP and solved using SeDuMi based on balanced state-space system realizations to improve the numerics. To decide whether $\boldsymbol{X}_\theta$ or $\boldsymbol{Y}_\theta$ should be held constant, the optimization problems were solved using all possible combinations of constant and parameter-dependent Lyapunov matrices, $\boldsymbol{X}_\theta$ and $\boldsymbol{Y}_\theta$. For all controllers, the smallest γ values were obtained by choosing $\boldsymbol{X}_\theta$ to be a constant matrix and $\boldsymbol{Y}_\theta$ to be dependent on θ.

The obtained γ values for the four optimization problems are shown in Table 1. For the PFTC, the ratio γ_f/γ_n was set to 2.8 in the optimization problem, which is a compromise made to achieve good performance in the normal situation and acceptable performance in the fault case. The tradeoff is illustrated in Fig. 9. The compromise was found by solving the PFTC for $\gamma_f/\gamma_n = 1$ and increasing this ratio until γ_n was only slightly improving (decreased) for increasing γ_f/γ_n.

From Table 1 it is seen that the nominal controller has the lowest γ_n value. This is expected as it is designed for only a subset of the models utilized in the design of each of the other three controllers. The robust controller is designed for the nominal case also, but captures the actual parameter variations in operation compared to the under-approximation utilized for the nominal controller. Therefore, γ_n is significantly higher for the robust controller. However, it is expected that the performance of the two controllers are closer when simulating the nonlinear system.

The fault-tolerant controllers appear to have larger values of γ in the fault-free case than for the nominal controller, which is expected since they are also accounting the fault. Additionally, there is a substantial difference between the γ values of the AFTC and PFTC, which was not the case for the AFTC and PFTC presented in [17], where a less severe fault was considered. These results indicate that a fault changing the system behavior significantly makes it very difficult for the PFTC to manage the normal and fault situations without degrading performance significantly. This claim was also supported by a design of the AFTC, where $\boldsymbol{X}_\theta$ was restricted to only depend on θ_{op} instead of depending on both θ_{op} and θ_f. In particular, γ_n was increased from 1.76 to 2.20. It was expected that the performance of the AFTC is superior to that of the PFTC.

5. Simulation results

The purpose of this section is to evaluate by simulation if the performance of the controllers resembles the γ values presented

in Table 1. Additionally, the different controllers should be compared to determine the impact of the different control strategies. The reference controller described in Section 3 was used as a frame of reference, being an approximation of a simplified wind turbine controller.

Simulations were conducted in MATLAB Simulink using the nonlinear model provided in Section 2 and the additive measurement noise from Section 2.6. The simulation model is nonlinear due the output model of the power in (8) and the nonlinear aerodynamics in (1) and (2) with the nonlinear mappings C_p and C_t. This implies that the model utilized for the controller design differs from the simulation model in all terms related to these nonlinear equations.

To compare the performance of the LPV controllers, simulations of duration 5000 s were conducted at wind speeds ranging from 13 m/s to 25 m/s to cover the entire design region. Simulations were conducted with and without including the fault in the pitch system to compare the performance of the controllers in case of low pressure in the hydraulic pitch system. Simulations at intermediate pressure levels are presented in the next subsection.

5.1. Validation of LPV controllers for the operational range and the rate bounds

To verify that the LPV controllers are able to operate in the entire parameter space and at the rate bounds, simulations were conducted where the wind turbine model is forced between the extremes of the parameter space used in the design. The first 125 s of the simulations are shown in Fig. 11, for both NC (green) and RC (magenta) designed for the nominal plant model and AFTC (blue) and PFTC (red) designed also to manage the fault in the pitch model.

From Fig. 11 it is concluded that the LPV controllers manage to operate in the entire parameter space and at the rate bounds. The right subplots reveal a significant difference between using the passive fault-tolerant controller or the active fault-tolerant controller, which is evident from the degraded tracking capability of the nominal generator speed reference of the PFTC (red) and from its larger pitch angle variations. Comparing operation in the fault-free and in the fault scenario shows that the AFTC (blue) has larger variations in the generator speed in the fault scenario (when $\theta_f = 1$).

5.2. Robustness analysis of robust controller

In order to verify that the robust controller meet the specification in the entire parameter space defined in Fig. 8, the parameter space was divided into a fine grid and the BMIs was tested in each point of the grid. Additionally, a nonlinear simulation of duration 5000 s was conducted to create Fig. 12, which shows that the wind turbine controlled by the robust controller operates in the entire parameter space defined in the design specification. Since the robust controller behaves satisfactory in the entire parameter space, it is concluded to satisfy the robustness requirements.

5.3. Comparison of controllers in the full load region

To demonstrate the feasibility of wind turbine LPV controllers, the nominal LPV controller was designed for the entire full load region and is compared to the reference controller in Fig. 13. The figure displays the first 150 s of the simulations and emphasizes that the nominal LPV controller (green) and the PI controller (gray) behave differently, due to their fundamental structural differences. Simulation results of the robust controller (magenta) are also shown in the figure and appear to have strong similarities to those

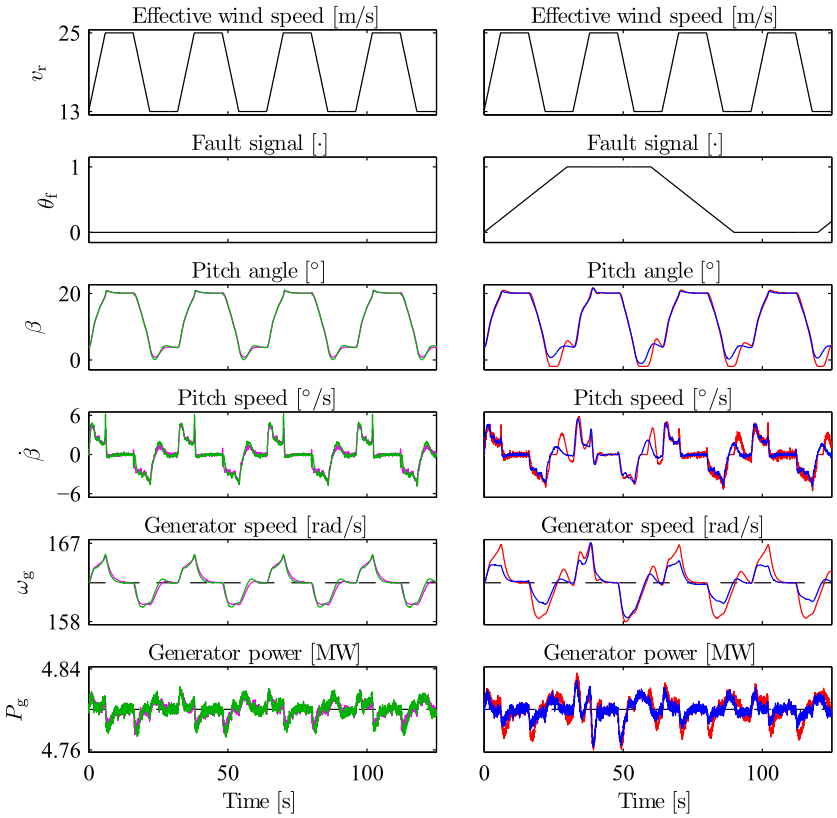

Fig. 11. Simulation results where NC (green), RC (magenta), AFTC (blue), and PFTC (red) are forced between the extremes of the operational range of the wind speed at the rate bounds. For the fault-tolerant controllers in the right column, the fault signal is changed between normal pressure ($\theta_f = 0$) and low pressure ($\theta_f = 1$) using the rate bounds. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

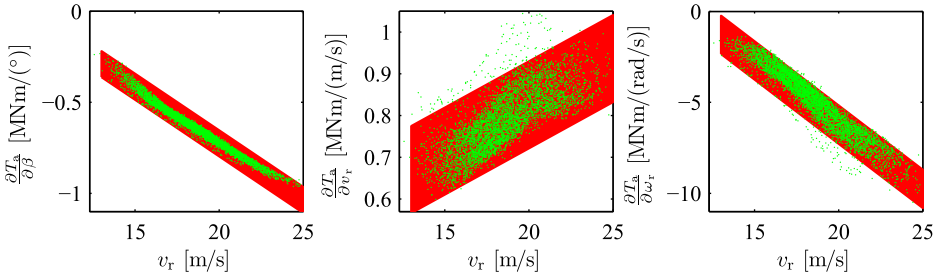

Fig. 12. Results of the robust controller (green) showing the partial derivatives of the aerodynamic torque in simulation. The partial derivatives of the aerodynamic torque are approximated by an affine uncertain descriptions (red). The robust controller has satisfactory performance in the entire parameter space. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

of the nominal controller, which is expected, since they are designed based on the same LPV plant model.

Even though the robust controller is designed for an uncertain model, it performs similar to the nominal controller, which appears from the list of performance measures in Table 2. This may be related to the fact that the nominal controller is designed for a model which has a simplification of the aerodynamic model, while the robust controller is designed for a model which captures more dynamics.

5.4. Comparison of controllers in the fault case

The simulation results of the fault-tolerant controllers are compared to the results of the reference controller in Fig. 14, showing a 1 min excerpt from the simulations. The AFTC (blue) and PFTC (red) are designed to manage low pressure in the pitch system, which is not the case for the reference controller (gray) that performs poorly in the fault case, showing oscillations in the control signal and increased tower accelerations.

Comparing the fault-tolerant controllers it is seen that the active fault-tolerant controller performs significantly better, since controller adaptation is offered based on the fault diagnosis signal. The performance measures in Table 2 show the performance of the

Table 2
Speed tracking errors and pitch actuator usage, normalized to the performance of the reference controller. The numbers in parentheses denote normalization with respect to the reference controller in the fault scenario. The performance measures are computed based on 5000 s simulations ranging between wind speeds of 13 m/s and 25 m/s.

Controller	$\int_0^t(\omega_{g,e}(\tau))^2 d\tau$	$\int_0^t \dot{\beta}^2(\tau) d\tau$
Normal situation		
NC	0.52	0.98
RC	0.53	0.96
AFTC	0.51	0.96
PFTC	1.09	1.02
Reference	1.00	1.00
Low hydraulic pressure		
NC	2.52 (0.75)	8.42 (0.73)
RC	1.54 (0.46)	4.76 (0.41)
AFTC	1.16 (0.34)	0.96 (0.08)
PFTC	1.96 (0.58)	1.22 (0.11)
Reference	3.38 (1.00)	11.52 (1.00)

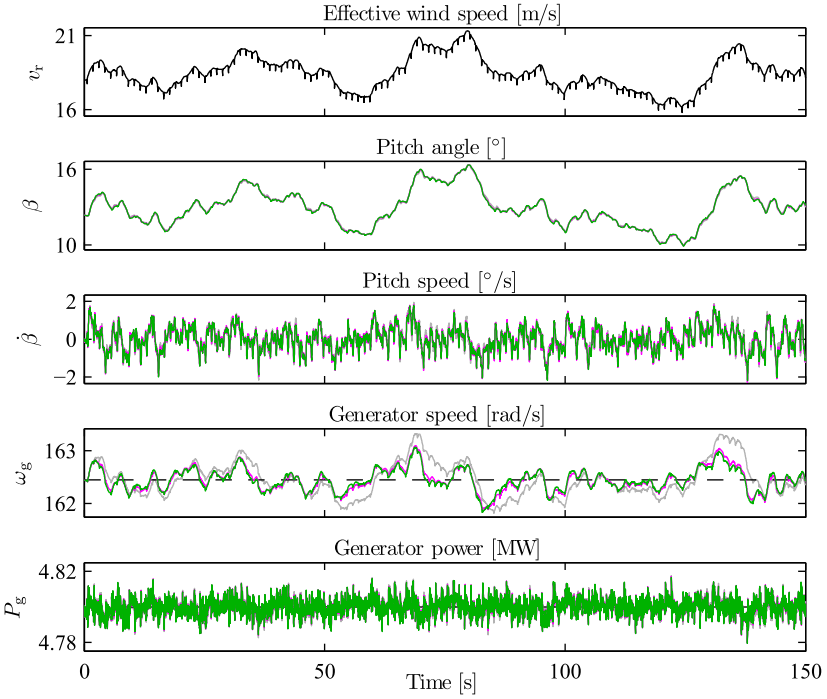

Fig. 13. Simulation results of the NC (green) and the RC (magenta), which can be compared to the operation of the reference controller (gray). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

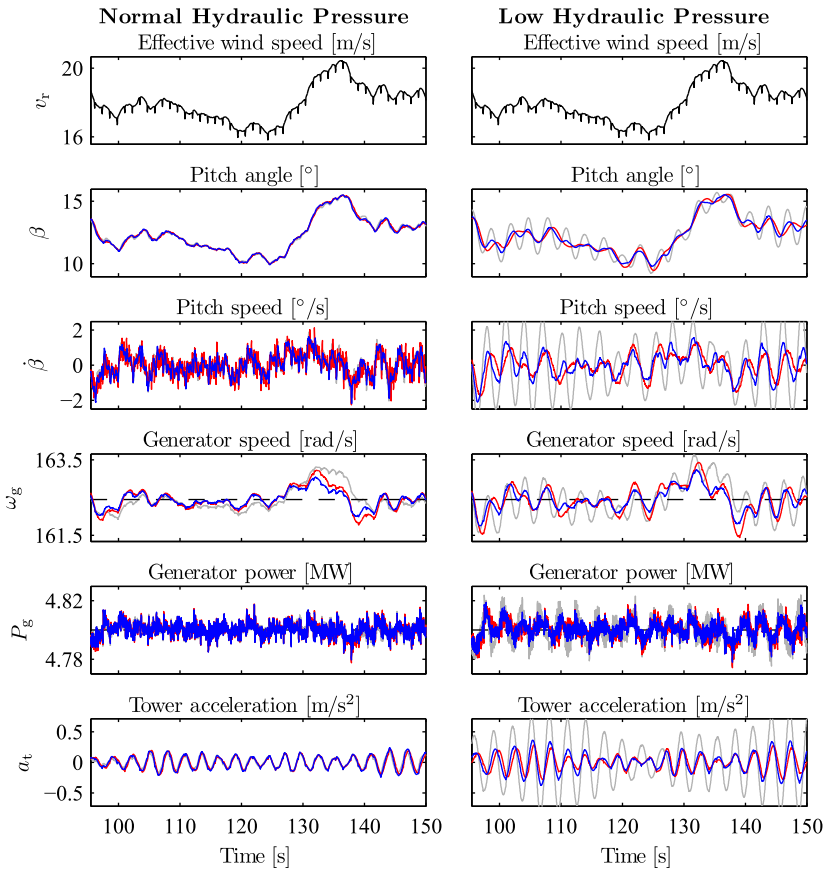

Fig. 14. Simulation results of the AFTC (blue) and the PFTC (red) conducted at both normal and low pressure in the hydraulic pitch system. The behaviors of the fault-tolerant controllers can be compared to the operation of the reference controller (gray). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

LPV controllers to be superior to the performance of the reference controller in the fault case. The pitch usage is significantly higher for NC and RC than for the fault-tolerant controllers, which is expected since the nominal controller and the robust controller do not include the pitch system fault in their design specifications. Among these two controllers, the robust controller has an advantage over the nominal controller as it utilizes an uncertain model, where the partial derivatives of the aerodynamic torque are allowed to change arbitrarily fast in the vertical direction of Fig. 8, which partly accounts for altered dynamics of the pitch model.

5.5. Performance measures

The performance measures obtained for the simulations presented in the previous subsections are stated in Table 2.

5.6. Discussion

Based on the simulation results it is concluded that the LPV controllers, except the PFTC, have almost the same performance in the normal situation and that the fault-tolerant controllers are superior in the fault situation. However, this does not completely resemble the a priori expectation based on the γ values presented in Table 1, where the robust controller has a much larger γ value than the nominal controller. This relates to the modeling used by the controllers, where the nominal controller utilizes an under-approximation of the nominal system model, due to the affine approximation of the partial derivatives of the aerodynamic torque, while the robust controller is designed for the actual parameter variations of the nominal system, due to the robustness added in the approximation of the partial derivatives of T_a. This result indicates that adding robustness in the control system to the expected parameter variations will not degrade the final result. Furthermore, if the controllers were applied to a real wind turbine, it is expected that the robust controller would have the biggest chance of performing satisfactory, as the robust design method includes dynamics which is not present in the model, as the uncertain parameters of the robust model are allowed to change arbitrarily fast and independent of each other. Similarly, the results in Table 2 show that the robust controller has superior performance to the nominal controller in the fault scenario, even though they have similar performance in the fault-free case.

The considered fault causing low hydraulic pressure in the pitch system has a significant influence on the system behavior, which is expected based on Fig. 3. Therefore, the AFTC performs significantly better than the PFTC, since the AFTC utilizes controller adaptation based on the fault diagnosis signal. This ensures better performance in the fault-free and fault situations by avoiding the conservatism introduced by the passive fault-tolerant design. The obtained simulation results fit well with the γ values presented in Table 1, where the large γ in the fault case for the passive fault-tolerant controller is required in order to achieve reasonable performance in the fault-free situation.

In [17] a similar wind turbine system is considered except that it includes high air content in the hydraulic oil as the pitch system fault; the two faults are compared in [9]. Here it is concluded that a passive and an active fault-tolerant controller have similar performance, because the fault has only a minor impact on the system behavior. For faults where the system dynamics are not changed significantly, passive fault-tolerant control may therefore be ap-

plied without any notable performance degradation; however, if the dynamics of the system are changed significantly, then active fault-tolerant control should be used to maintain a satisfactory nominal performance.

6. Conclusion

This paper addresses the design of four LPV controllers for a wind turbine operating in the full load region. The designed controllers demonstrate that it is possible to add robustness and fault-tolerant capabilities to a nominal LPV controller by utilizing a unified LMI-based design method. All controllers handle the parameter variations along the nominal operating trajectory. The robust controller also provides robustness towards parameter uncertainties in the aerodynamic model, while the fault-tolerant controllers handle the parameter variations introduced by a fault in the hydraulic pitch system. The paper includes both an active fault-tolerant controller and a passive fault-tolerant controller, being the two possible choices.

The method is based on parameter-dependent Lyapunov functions, which reduces conservativeness of control for systems with rate bounds, which is the case in this work. In case of abrupt component failures constant Lyapunov functions are required, and can be applied although they introduce more conservatism. In the considered case, the parameter variations are approximated using an affine description to simplify the solution. If an affine approximation cannot be justified, then uncertain parameters can be introduced to cover model nonlinearities, which though may be conservative for highly nonlinear systems; the only other alternative is to apply gridding in the parameter space.

Simulations show that the LPV controllers are superior to a reference controller designed using classical methods. Furthermore, it is shown that robustness to the expected parameter variations can be added to the controller design without sacrificing the performance gap compared to the reference controller. From the simulation results it is apparent that faults should be taken into account in the controller design procedure, since the controllers designed for the nominal system start oscillating when the fault is introduced. Comparing the fault-tolerant controllers, it is seen that the active fault-tolerant controller performs significantly better than the passive fault-tolerant controller, since controller adaptation is offered based on the fault diagnosis signal. This enables less conservatism resulting in better performance.

In relation to the numerics involved in synthesizing the four controllers, it is easier to solve the optimization problems for the NC and AFTC than for the RC and PFTC which have uncertain parameters, as the optimization problems are LMI-based and therefore can be solved using convex optimization methods. However, using the presented design procedure it is possible to solve the BMIs and achieve controllers with a good performance, even though there are no guarantees for obtaining a global minimum in the optimization problem.

In general, robust control is capable of securing stability and satisfactory performance despite model uncertainties. Simulations show that the nominal and the robust controllers have the same performance for the nonlinear model; however, as additional guarantees are provided by the robust controller design, this method is preferred. With respect to fault-tolerant control, an AFTC should be used on systems for which a fault diagnosis system can be designed to be sufficiently fast with a low risk of making false decisions. If a fault changes the system behavior significantly, then an AFTC should also be applied because controller adaptation will have a large impact on performance. A PFTC should be favored when faults are difficult to diagnose, when it has only a small impact on system performance, or there is zero tolerance for false decisions in the fault diagnosis system.

Acknowledgments

The authors would like to thank Fabiano Daher Adegas for his help with improving the conditioning of the LMIs in the controller design problems, and Kim Billesø Danielsen for proofreading the paper.

Appendix A. Model parameters

The following parameters represent a realistic but fictitious variable-speed, variable-pitch wind turbine. The nominal power is 4.8 MW and the rotor diameter is 115 m.

Aerodynamic model

$A = 10387\ \mathrm{m}^2$ and standard air density $\rho = 1.225\ \mathrm{kg/m}^3$. The C_p mapping is illustrated in Fig. 2 and is available from the model in [23], and the C_t mapping can be found in [9].

Drive train model

$B_r = 27.8\ \mathrm{kNm/(rad/s)}$, $B_g = 3.034\ \mathrm{Nm/(rad/s)}$, $B_{dt} = 945\ \mathrm{kNm/(rad/s)}$, $J_r = 55 \cdot 10^6\ \mathrm{kg\,m}^2$, $J_g = 390\ \mathrm{kg\,m}^2$, $K_{dt} = 2.7\ \mathrm{GNm/rad}$, and $N_g = 95$.

Tower model

$B_t = 66.7\ \mathrm{N/(m/s)}$, $K_t = 2.55\ \mathrm{MN/m}$, and $M_t = 484$ t.

Pitch system model including fault model

$t_d = 10$ ms, $\beta \in [-1.9^\circ, 40^\circ]$, and $\dot{\beta} \in [-10^\circ/\mathrm{s},\ 10^\circ/\mathrm{s}]$.

- Nominal values (normal pressure in hydraulic pitch system): $\omega_{n,0} = 11.11$ rad/s and $\zeta_0 = 0.6$ rad/s.
- Fault values (low pressure in hydraulic pitch system): $\omega_{n,lp} = 3.42$ rad/s and $\zeta_{lp} = 0.9$ rad/s.

Generator model

$\eta_g = 0.92$.

Converter model

$t_{g,d} = 20$ ms, $\tau_g = 10$ ms, $T_g \in [0\ \mathrm{Nm}, 35.3\ \mathrm{kNm}]$, and $\dot{T}_g \in [-50\ \mathrm{MNm/s}, 50\ \mathrm{MNm/s}]$.

Appendix B. Reference controller

This section includes the parameters of the reference controller.

PI-controllers

Speed controller 1: $K = -6.89^\circ/(\mathrm{rad/s})$, $T_i = 25$ s.
Speed controller 2: $K = -2.95^\circ/(\mathrm{rad/s})$, $T_i = 6.02$ s.
Power controller: $K = 447 \cdot 10^{-6}\ \mathrm{Nm/W}$, $T_i = 0.031$ s.

Drive train damping

$\tau_d = t_{g,d} = 20$ ms, $K_d = 2500\ \mathrm{Nm/(rad/s)}$, and $\zeta_d = 0.25$.

References

[1] Østergaard KZ. Robust, gain-scheduled control of wind turbines. PhD thesis, Aalborg University; 2008.
[2] Hammerum K. A fatigue approach to wind turbine control. Master's thesis, Technical University of Denmark; 2006.
[3] Sloth C, Esbensen T, Niss MOK, Stoustrup J, Odgaard PF. Robust LMI-based control of wind turbines with parametric uncertainties. In: Proceedings of the 3rd IEEE multi-conference on systems and control. Saint Petersburg, Russia; 2009. p. 776–81.
[4] Niss MOK, Esbensen T, Sloth C, Stoustrup J, Odgaard PF. A Youla–Kucera approach to gain-scheduling with application to wind turbine control. In: Proceedings of the 3rd IEEE multi-conference on systems and control. Saint Petersburg, Russia; 2009. p. 1489–94.
[5] Laks JH, Pao LY, Wright AD. Control of wind turbines: past, present, and future. In: Proceedings of the 2009 American control conference. St. Louis, MO, USA; 2009. p. 2096–103.
[6] Bianchi FD, Battista HD, Mantz RJ. Wind turbine control systems: principles, modelling and gain scheduling design. Springer; 2007, ISBN 1-84628-492-9.
[7] Niemann H, Stoustrup J. An architecture for fault tolerant controllers. International Journal of Control 2005;78(14):1091–110.
[8] Stoustrup J, Blondel V. Fault tolerant control: a simultaneous stabilization result. IEEE Transactions on Automatic Control 2004;49(2):305–10.
[9] Esbensen T, Sloth C. Fault Diagnosis and Fault-Tolerant Control of Wind Turbines. Master's thesis, Aalborg University; 2009.
[10] Odgaard PF, Stoustrup J, Kinnaert M. Fault tolerant control of wind turbines – a benchmark model. In: Proceedings of the 7th IFAC symposium on fault detection, supervision and safety of technical processes. Barcelona, Spain; 2009. p. 155–60.
[11] Dolan DSL, Lehn PW. Simulation model of wind turbine 3p torque oscillations due to wind shear and tower shadow. IEEE Transactions on Energy Conversion 2006;21(3):717–24.
[12] Aalborg University and Risø National Laboratory. Wind Turbine Blockset; 2005.
[13] Apkarian P, Adams R. Advanced gain-scheduling techniques for uncertain systems. IEEE Transactions on Control Systems Technology 1998;6(1):21–32.
[14] Packard A. Gain scheduling via linear fractional transformations. Systems & Control Letters 1994;22(2):79–92.
[15] Gahinet P, Apkarian P, Chilali M. Affine parameter-dependent Lyapunov functions and real parametric uncertainty. IEEE Transactions on Automatic Control 1996;41(3):436–42.
[16] Wu F, Yang XH, Packard A, Becker G. Induced l_2-norm control for lpv systems with bounded parameter variation rates. International Journal of Robust and Nonlinear Control 1996;6(9–10):983–98.
[17] Sloth C, Esbensen T, Stoustrup J. Active and passive fault-tolerant LPV control of wind turbines. In: Proceedings of the 2010 American control conference. Baltimore, USA; 2010. p. 4640–6.
[18] Østergaard K, Brath P, Stoustrup J. Estimation of effective wind speed. Journal of Physics: Conference Series 2007;75(1):1–9.
[19] Hallouzi R. Multiple-model based diagnosis for adaptive fault-tolerant control. PhD thesis, Delft University of Technology; 2008.
[20] Apkarian P, Gahinet P, Becker G. Self-scheduled H_∞ control of linear parameter-varying systems: a design example. Automatica 1995;31(9):1251–61.
[21] Jabbari F. Output feedback controllers for systems with structured uncertainties. IEEE Transactions on Automatic Control 1997;42(5):715–9.
[22] Chilali M, Gahinet P. H_∞ design with pole placement constraints: an LMI approach. IEEE Transactions on Automatic Control 1996;41(3):358–67.
[23] kk-electronic a/s. Fault tolerant control of wind turbines a benchmark model; 2010. <www.kk-electronic.dk/Default.aspx?ID=9338>.