ECN

Energy research Centre of the Netherlands

Wind Turbine Extreme Gust Control

Recognition and Control of extreme operation gusts and wind direction changes

Stoyan Kanev

Tim van Engelen

ECN-E–08-069

Abstract

This report presents the research activities and achieved results on extreme event recognition (EER) and control (EEC). This work has been performed within the framework of WP3 of the SenterNovem project *“Sustainable Control (SusCon). A new approach to operate wind turbines”* with project number EOSLT02013.

An extreme wind gust with direction change can lead to large loads on the turbine (causing fatigue) and unnecessary turbine shut-downs by the supervisory system due to rotor overspeed. The proposed EER algorithm is based on a nonlinear observer (extended Kalman filter) that estimates the oblique wind inflow angle and the blade effective wind speed signals, which are then used by a detection algorithm (CUSUM test) to recognize extreme events. The nonlinear observer requires that blade root bending moments measurements (in-plane and out-of-plane) are available. Once an extreme event is detected, an EEC algorithm is activated that (i) tries to prevent the rotor speed from exceeding the overspeed limit by fast collective blade pitching, and (ii) reduces 1p blade loads by means of individual pitch control algorithm, designed in an $\mathcal{H}_\infty$ optimal control setting. The method is demonstrated on a complex nonlinear test turbine model.

Acknowledgment

Our colleagues Koert Lindenburg and Jan Schuurmans are acknowledged for the useful discussions and suggestions.

Keywords

wind speed estimation, wind direction estimation, blade-effective wind speeds, extreme event recognition, extreme event control, load reduction.

Contents

List of Symbols

The following symbols (with SI dimensions) are used in the text (see also Figure 1 for visualization of some wind speed definitions):

c_A	cord length of blade element A,
C_L, C_D, C_M	lift, drag, and pitch-wise torque coefficients,
$F_p^{A,b}$	Prandtl’s tip and root correction term,
M_x^b, M_z^b	lead-wise (in-plane) and flap-wise (out-of-plane) blade b root bending moment,
M_k	$(= [M_x^1, M_x^2, M_x^3, M_z^1, M_z^2, M_z^3]^T)$ vector of blade root bending moments,
P_k	state covariance matrix in the extended Kalman filter,
$q_t^{A,b}$	aerodynamic pitch-wise moment (nose-down positive) of element A of blade b,
$q_{f,n}^{A,b}$, $q_{f,l}^{A,b}$	aerodynamic forces in normal and leadwise direction for element A of blade b,
$q_{n,lift}^A$	annulus-averaged lift component of the aerodynamic forces in normal direction,
R	rotor radius,
r_A	distance from rotor center to the middle of annulus A.
δr_A	breadth of annulus A,
r_{root}	blade root radius,
S_A	$(= 2\pi r_A \delta r_A)$ area of rotor annulus A
T_g	generator torque reference (output of controller),
T_s	sampling time at which the turbine model runs,
T_s^{ctr}	sampling time at which the control algorithms run,
$\bar{U}$	mean undisturbed wind speed in the longitudinal wind field direction,
$\bar{U}_{ax}$, $\bar{U}_{yw}$, $\bar{U}_{tlt}$	axial, yaw-oriented and tilt-oriented components of $\bar{U}$
$\bar{U}_i^A$, $\bar{V}_i^A$	equilibrium axial and tangential induction wind speeds,
δU_i^A	dynamic term on the axial induction wind speed,
$\delta U_{i,corr}^{A,b}$	Glauert’s correction term to U_i^A for oblique inflow,
$U_i^{A_{2/3}}$	$(= \bar{U}_i^{A_{2/3}} + \delta U_i^{A_{2/3}})$ axial induction wind speed of annulus at $2/3R$
$U_{tr}^{A,b}$	transport velocity of the wake,
$U_{tr,F}^A$	annulus-averaged transport velocity including Prandtl’s tip/root correction term,
u_b	blade b effective wind speed,
$\bar{V}_n^A$, $\bar{V}_l^A$	equilibrium normal and lead-wise effective wind speed at blade element A,
$\delta V_n^{A,b}$, $\delta V_l^{A,b}$	normal and lead-wise effective wind speed variation at element A of blade b,
x	state of the (reduced) SDS model,
x^a	augmented state, consisting of x, $[u_1, u_2, u_3]^T$ and β,
$\alpha^{A,b}$	angle of attack of element A of blade b,
β	additional (to $\bar{\phi}_{yw}$) yaw misalignment angle for modeling wind direction change,
ζ	rotor coning angle,
θ_b	pitch angle reference for blade b (output of controller),
ρ	air density,
$\delta\bar{\phi}^A$	angle of attack correction for blade element A due to rotor coning,
$\phi^{A,b}$	pitch angle of element A of blade b,
ϕ^b	$(= \phi^{1,b})$ pitch angle of blade b, measured at the blade root,
$\bar{\phi}_{yw}$, $\bar{\phi}_{tlt}$	equilibrium yaw and tilt angles of the wind speed $\bar{U}$ (see Figure 1),
ψ^b	azimuth angle of blade b,
$\psi = \psi_1$	rotor azimuth angle,
$\delta\psi$	azimuth offset angle due to oblique inflow orientation,
Ω	rotor speed,
Ω^f	filtered rotor speed,

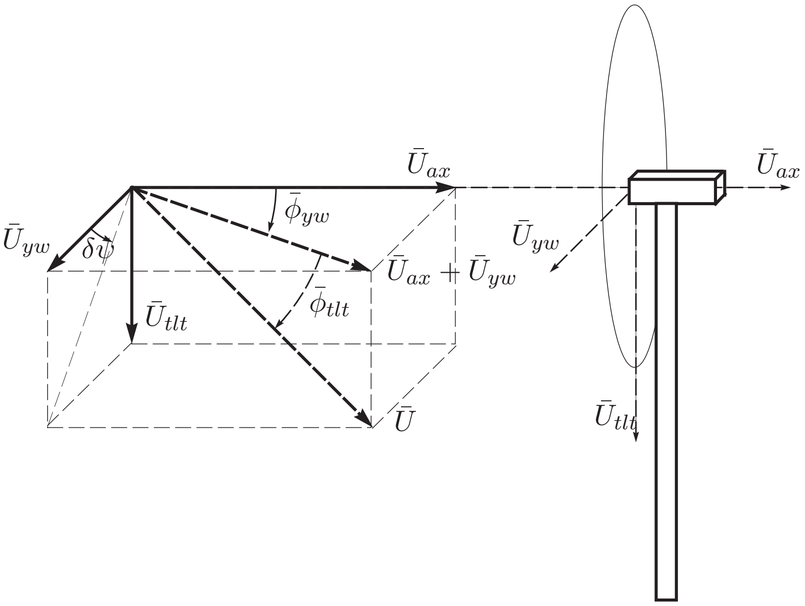


Figure 1: Definitions of tilt $\bar{U}_{tlt}$, yaw $\bar{U}_{yw}$ and axial $\bar{U}_{ax}$ oriented components of the equilibrium wind vector $\bar{U}$, and yaw $\bar{\phi}_{yw}$ and tilt $\bar{\phi}_{tlt}$ angles.

1 Introduction

Extreme wind conditions, such as wind gusts and/or wind direction changes, can lead to very large turbine loads causing fatigue, automatic shut-downs or even damage to some turbine components. Such effects could be circumvented by means of timely recognition of the extreme event (*extreme event recognition*), followed by a promptly and proper control system reaction (*extreme event control*). In this report the extreme event recognition (EER) is performed by means of estimating the oblique inflow angle (yaw misalignment) together with blade-effective wind speed signals from measurements on the flapwise (out-of-plane) and leadwise (in-plane) bending moments in the blade roots. These estimates are used to recognize extreme events (wind gusts and/or wind direction changes), which activates an extreme event control (EEC) algorithm. The EEC has on the one hand the purpose of preventing rotor overspeed (which can trigger complete turbine shutdown by the supervisory system) by collectively pitching the blades toward feather, and on the other hand to reduce 1p (once per revolution) blade loads by individually pitching the blades.

The problem of *rotor-effective* wind speed estimation has been addressed in the literature on several occasions, where the usual approach is to estimate the aerodynamic torque on the rotor $T_a(u)$, which is subsequently inverted to obtain the rotor-uniform wind speed u. The estimation of T_a is done either by neglecting the rotor dynamics and using the static power-wind curve (Thiringer and Petersson, 2005; Ma et al., 1995), or by considering a simple first-order model of the rotor dynamics, i.e. neglecting shaft torsion, (van der Hooft and van Engelen, 2004; Kodama and Matsuzaka, 2000; Sbarbaro and Peña, 2000). Recently, somewhat more advanced models have been used, including first shaft torsion mode to the rotor dynamics (Østergaard et al., 2007). In estimating the aerodynamic torque, the majority of these methods rely on the computation of the time-derivative of the rotor speed measurement, and are as such very sensitive to measurement noise as well as to unmodelled higher order dynamics such as tower sidewards motion and collective blade lead-lag motion. To avoid this, appropriate filtering of the rotor speed is necessary, which inevitably introduces time delay and, hence, sacrifices the performance of the wind estimator. More advanced methods have, though, also been studied, including extended Kalman filter (Ma et al., 1995), linear Kalman filter in combination with T_a-tracking control loop (Østergaard et al., 2007), or augmented-state nonlinear filters (Sbarbaro and Peña, 2000). Still, all these publications have several things in common: they all assume one single rotor-effective wind speed signal, no yaw misalignment, a rigid rotor and tower, and use equilibrium-wake aerodynamics based on static power-wind curves.

To the best of the author's knowledge there has been no publication on simultaneous estimation of *blade*-effective wind speeds and yaw misalignment angle, which is in the basis of the EER algorithm developed in this report. More specifically, an augmented state extended Kalman filter (EKF) is utilized, based on a nonlinear wind turbine model. This model consists of a linear structural dynamics module (SDM) on which aerodynamic forces and torques are acting as com-

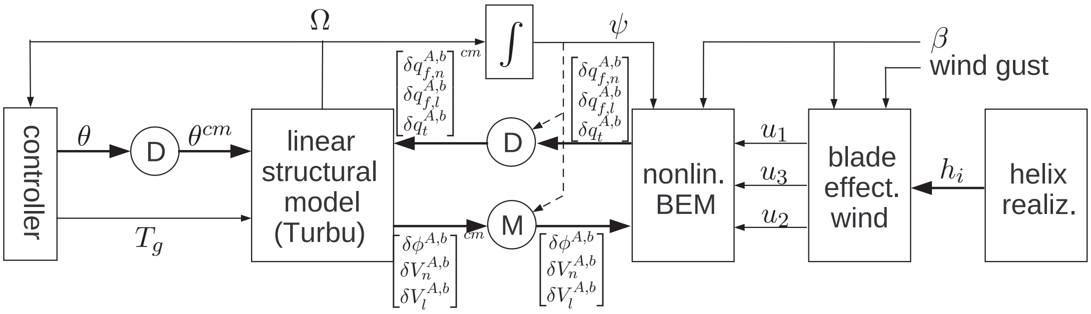

Ω
∫
ψ
β
wind gust
controller
θ
D
θ^{cm}
linear structural model (Turbu)
T_g
$\begin{bmatrix}\delta q_{f,n}^{A,b}\\ \delta q_{f,l}^{A,b}\\ \delta q_t^{A,b}\end{bmatrix}^{cm}$
D
$\begin{bmatrix}\delta q_{f,n}^{A,b}\\ \delta q_{f,l}^{A,b}\\ \delta q_t^{A,b}\end{bmatrix}$
M
$\begin{bmatrix}\delta \phi^{A,b}\\ \delta V_n^{A,b}\\ \delta V_l^{A,b}\end{bmatrix}^{cm}$
$\begin{bmatrix}\delta \phi^{A,b}\\ \delta V_n^{A,b}\\ \delta V_l^{A,b}\end{bmatrix}$
nonlin. BEM
u_1
u_3
u_2
blade effect. wind
h_i
helix realiz.

Figure 2: Turbine simulation scheme

puted by a nonlinear aerodynamic conversion module (ACM), driven by realistic blade-effective wind speed signals. Compared to the model used in the Kalman filter, a model of an even higher complexity is used for simulation and analysis, the main components of which are given in block-schematic form in Figure 2 (in which the physical meaning of the signals is described later on). These components are:

- 40-th order linearized structural dynamics model (SDM), obtained using the software TURBU (van Engelen, 2007), with degrees of freedom in tower foundation, blade flanges and drive train, and including pitch actuator dynamics,
- nonlinear aerodynamic conversion module (ACM) based on blade element momentum (BEM) theory, including
    - dynamic wake effects as modeled by the *ECN Differential Equation Model* (Snel and Schepers, 1994),
    - Glauert's azimuth-dependent correction term for the axial induction speed in case of oblique inflow (van der Hooft et al., 2007),
    - correction on the angle of attack due to rotor coning, as implemented in the nonlinear aero-elastic wind turbine simulation tool PHATAS (Lindenburg and Schepers, 2001),
- linear blade pitch controller regulating the filtered generator speed at its rated level (when operating at above-rated conditions), and consisting of a PI-controller in series with low-pass filter at the 3P blade frequency, notch filter at the first tower sideward frequency, and notch filter at the first collective lead-lag frequency,
- nonlinear generator torque controller based on static optimal-λ QN-curve at below rated conditions and *constant power* production above-rated, operating on the filtered generator speed signal (same three filters used as in pitch controller),
- additional azimuth-dependent nonlinearities arising from the Coleman transformations between the fixed reference frame (in which the input/output signals of the SDM are defined) and the rotating reference frame (in which the signals of the ACM are defined), see blocks M (modulation) and D (demodulation) in Figure 2,
- realistic blade effective wind speed signals are generated based on the helix approximation concept, including both a deterministic term for modeling wind shear, tower shadow, tilt and yaw misalignment, wind gust, and a stochastic term that models blade-effective turbulence.

The EKF uses a simplified model in which the structural dynamics model is reduced to order 20, and the ADM model excludes dynamic wake effects, as well as the effects of the structural dynamics onto the aerodynamics, i.e. the effects of the vibration and deformation of the blades and the tower onto the apparent wind speeds are neglected (the leadwise speeds of the blade elements resulting from the rotation of the rotor is, of course, not neglected, only the variations around these speeds).

Based on the blade-effective wind speeds and oblique inflow angle, estimated by the EKF, an extreme event detection mechanism is used, consisting of a cumulative sum (CUSUM) test that detects (significant) changes in the mean value of the estimated signals. Once the extreme event flag is raised by the CUSUM test, an EEC algorithm is activated that consists of two components. The first one is a rotor overspeed prevention algorithm that immediately starts pitching the blades to feather with the maximally allowed pitch speed, and at the same time sets the reference generator torque equal to its rated value. This action has the purpose to prevent rotor overspeed in order to avoid a possibly unnecessary turbine shutdown by the supervisory system. The conventional power control is turned back on when either the (filtered) rotor speed begins decreasing, or the pitch angles have reached a suitably defined reference value, which is a function of the axial component of the (estimated) wind speed. The last one is computed off-line under the assumption of rated rotor speed and rated generator torque. The process of switching the conventional control algorithm back on is performed in a bumpless manner by means of proper controller state re-initialization. The second component of the EEC consists of an individual pitch control (IPC) algorithm aiming at the reduction of 1p blade loads, which are rather large under oblique inflow conditions. A modern optimal-$\mathcal{H}_\infty$ control methodology is used for the design of the IPC. This loads reduction control should be only activated after the rotor overspeed prevention system is deactivated, as their simultaneous activity would require blade pitch speeds exceeding the maximal allowable speed. In fact, the IPC could, principally, be let working even when there is no extreme event, although the resulting continuous cyclic blade pitching might be undesirable. In the implementation in this report the IPC is only active whenever the estimated oblique inflow angle is larger (in absolute value) than 10^o.

The report is organized as follows. The next section explains the notation used throughout the report, as well as the physical meaning of the used variables. Section 3 describes the structure and the main components of the turbine simulation model. The algorithm for detection of extreme events in developed in Section 4, while extreme event control is the topic of Section 5. The complete EER-EEC method is tested in simulations in Section 6. The report is concluded in Section 7 with some concluding remarks.

2 Notation

For a scalar or vector variable v, $\bar{v}$ denotes its equilibrium or mean value, while $\delta v = v - \bar{v}$ is called the (current) variation around the equilibrium value. An superscript cm, as in v^{cm}, means that the variable is defined in multi-blade coordinates as obtained by performing a Coleman demodulation (see Section 4.1) of the signal v (v being defined in the rotating reference frame). Subscripts/subscripts b and A, as in $U_n^{A,b}$, denote the number of the blade ($b = 1, 2, 3$) and the number of the blade element ($A = 1, 2, \ldots, N_{ann}$) for which the variable is defined. For simplicity of notation it is assumed in the ADM that the number of blade elements is equal to the number of annuli, and that the length of the A-th blade element is equal to the breadth of annulus A. The operation $A \otimes B$ denotes the Kronecker product between A and B, while $\mathrm{vec}(A)$ stacks the columns of the matrix A below each other into one vector. The operator $\oplus$ represents the direct sum of matrices, i.e. $A \oplus B = \mathrm{blockdiag}(A, B)$. The n-by-n identity matrix is denoted as I_n, and $\delta_{b,i}$ is the Kronecker delta function.

3 Turbine Simulation Model

The turbine simulation model represents a typical 3-bladed horizontal axis wind turbine (HAWT). The model consists of an integration of several blocks, as sketched on Figure 2. These blocks are explained in more detail in the following subsections.

3.1 Structural dynamics system (SDM)

The SDM block consists of a linearized model, obtained with the software TURBU (van Engelen, 2007). The model assumes rigid blades and tower, but contains degrees of freedom in the blade flanges, in the tower foundation, in the rotor shaft, and includes the pitch actuator dynamics. Although the blades are considered rigid, there are $N_{ann} = 14$ blade elements per blade, allowing for a better representation of the aerodynamic forces, as computed from the ADM block, described in Section 3.3. The model (see Figure 2) has:

- 40 states: positions and speeds in 3 directions for the three blade flange elements and the tower bottom element, rotational position and speed for the two drive-train elements, and 4 states per blade for modeling the servo-pitch actuators at the three blades (all states defined in multi-blade coordinates, see Section 4.1),
- 130 inputs: 3 reference blade pitch angles θ^{cm}, one reference generator torque T_g, $3N_{ann}$ blade element torques q_t^{cm}, $3N_{ann}$ normal forces $q_{f,n}^{cm}$ and $3N_{ann}$ leadwise forces $q_{f,l}^{cm}$, all in multi-blade coordinates, and
- 133 outputs: rotor speed Ω, 3 blade root out-of-plane bending moments M_z^{cm}, 3 blade root in-plane bending moments M_x^{cm}, $3N_{ann}$ blade element pitch angles $(\delta\phi^{A,b})^{cm}$, $3N_{ann}$ normal velocities $(\delta V_n^{A,b})^{cm}$ and $3N_{ann}$ leadwise velocities $(\delta V_l^{A,b})^{cm}$, also in multi-blade coordinates.

The inputs θ^{cm} and T_g are controlled inputs, the outputs Ω, M_z^{cm}, and M_x^{cm} are assumed measured, and the remaining inputs and outputs are used for interconnecting the SDM with the ADM.

3.2 Wind generation

The generated blade effective wind speeds u_b have two components: a deterministic component which is the same for all blades and is used to represent wind gusts, wind shear and tower shadow, and a stochastic turbulence component, which is computed on the basis of the helix interpolation algorithm, described Appendix C. These blade-effective wind speeds are computed in such a way that the resulting flapwise blade root bending moments approximate (in terms of spectrum) those arising from a three-dimensional wind field turbulence. The blade effective wind speed signals are defined in longitudinal wind field direction (i.e. parallel to the undisturbed wind vector $\bar{U}$). In addition to that, an oblique inflow angle β is generated by the wind generation module, which represents yawed flow.

3.3 Aerodynamic module (ADM)

The ADM consists is summarized in an algorithmic form as follows:

Algorithm 3.1 (ADM)

Given
Equilibrium values and parameters from TURBU*:* $\bar{U}_{ax}$, $\bar{U}_{yw}$, $\bar{U}_{tlt}$, $\bar{U}_i^A$, $\bar{V}_i^A$, $\bar{V}_n^A$, $\bar{V}_l^A$, $\bar{\phi}^{A,b}$, $\bar{q}_{f,n}^{A,b}$, $\bar{q}_{f,l}^{A,b}$, $\bar{q}_t^{A,b}$, r_A, c_A, S_A, δr_A, R, ζ, ρ, $C_L(\alpha)$, $C_D(\alpha)$, $C_M(\alpha)$.
From SDM: ψ, $\delta\phi^{A,b}$, $\delta V_n^{A,b}$, $\delta V_l^{A,b}$
From wind module: β, u_b
From ADM at previous time instant: δU_i^A

Step 1 *Compute transformation matrix $T(\beta)$ mapping longitudinal wind field direction (direction of $\bar{U}$) to the axial, yaw and tilt direction (see Figure 1):*

$$\begin{aligned}
&\bar{U} = \sqrt{\bar{U}_{ax}^2 + \bar{U}_{yw}^2 + \bar{U}_{tlt}^2},\\
&\bar{\phi}_{yw} = \arctan\left(\left|\frac{\bar{U}_{yw}}{\bar{U}_{ax}}\right|\right) \operatorname{sgn}\left(\bar{U}_{yw}\right),\\
&\bar{\phi}_{tlt} = \arccos\left(\left|\frac{\sqrt{\bar{U}_{ax}^2+\bar{U}_{yw}^2}}{\bar{U}}\right|\right) \operatorname{sgn}\left(\bar{U}_{tlt}\right),\\
&T(\beta) = \begin{bmatrix} T_1(\beta) \\ T_2(\beta) \\ T_3 \end{bmatrix} = \begin{bmatrix} \cos(\bar{\phi}_{tlt})\cos(\bar{\phi}_{yw}+\beta) \\ \cos(\bar{\phi}_{tlt})\sin(\bar{\phi}_{yw}+\beta) \\ \sin\bar{\phi}_{tlt} \end{bmatrix}
\end{aligned} \tag{1}$$

Step 2 *Compute undisturbed wind speeds, including current blade-effective wind speed variations u_b (containing turbulence and wind gusts):*

$$\begin{aligned}
&\begin{bmatrix} U_{ax}^{\beta} \\ U_{yw}^{\beta} \\ U_{tlt}^{\beta} \end{bmatrix} = T(\beta)\bar{U},\\
&\psi^b = \psi + \tfrac{2(b-1)\pi}{3},\ b = 1, 2, 3.\\
&u^R = \tfrac{1}{3}\textstyle\sum_{b=1}^{3} u_b,\\
&\delta u_b = u_b - u^R,\\
&\begin{bmatrix} U_{ax}^{\beta,gust} \\ U_{yw}^{\beta,gust} \\ U_{tlt}^{\beta,gust} \end{bmatrix} = \begin{bmatrix} U_{ax}^{\beta} \\ U_{yw}^{\beta} \\ U_{tlt}^{\beta} \end{bmatrix} + T(\beta)u^R,
\end{aligned} \tag{2}$$

Step 3 *Compute Glauert's correction $\delta U_{i,corr}^{A,b}$ to the axial induction speed*

$$\begin{aligned}
&\delta\psi = \begin{cases}
\frac{\pi}{2}\operatorname{sgn}\left(U_{tlt}^{\beta,gust}\right), & \textit{if } U_{yw}^{\beta,gust} = 0.\\
\arctan\left(\left|\frac{U_{tlt}^{\beta,gust}}{U_{yw}^{\beta,gust}}\right|\right), & \textit{if } (U_{yw}^{\beta,gust} > 0)\&(U_{tlt}^{\beta,gust} \geq 0).\\
\pi - \arctan\left(\left|\frac{U_{tlt}^{\beta,gust}}{U_{yw}^{\beta,gust}}\right|\right), & \textit{if } (U_{yw}^{\beta,gust} < 0)\&(U_{tlt}^{\beta,gust} \geq 0).\\
\pi + \arctan\left(\left|\frac{U_{tlt}^{\beta,gust}}{U_{yw}^{\beta,gust}}\right|\right), & \textit{if } (U_{yw}^{\beta,gust} < 0)\&(U_{tlt}^{\beta,gust} < 0).\\
2\pi - \arctan\left(\left|\frac{U_{tlt}^{\beta,gust}}{U_{yw}^{\beta,gust}}\right|\right), & \textit{if } (U_{yw}^{\beta,gust} > 0)\&(U_{tlt}^{\beta,gust} < 0).
\end{cases}\\
&\delta U_{i,corr}^{A,b} = \frac{15\pi}{64R} r_A \tan\left(\frac{\arctan\left(\frac{\sqrt{\left(U_{yw}^{\beta,gust}\right)^2+\left(U_{tlt}^{\beta,gust}\right)^2}}{U_{ax}^{\beta,gust}-U_i^{A_{2/3}}}\right)}{2}\right)\cos(\psi^b - \delta\psi)U_i^{A_{2/3}}
\end{aligned} \tag{3}$$

Step 4 *Compute setting angles of blade elements $\phi^{A,b}$, including angle of attack correction*

$\delta\phi^{A,R}$ due to rotor coning (ζ)

$$\begin{aligned}
&\delta\phi^{A,R} = \tfrac{c_A}{2r_A}\sin\zeta,\\
&\phi^{A,b} = \bar{\phi}^{A,b} - \delta\phi^{A,R} + \delta\phi^{A,b}.
\end{aligned} \tag{4}$$

Step 5 *Compute normal $U_n^{A,b}$ and leadwise $U_l^{A,b}$ effective wind speeds and angle of attacks per blade element:*

$$\begin{aligned}
&\delta U_{ni}^{A,b} = \delta U_i^A - \delta U_{i,corr}^{A,b}\\
&U_n^{A,R} = U_{ax}^{\beta,gust} - \bar{U}_i^A - \bar{V}_n^A,\\
&U_l^{A,R} = -(\bar{V}_l^A + \bar{V}_i^A)\\
&\begin{bmatrix}\delta u_{ax}^b\\ \delta u_{yw}^b\\ \delta u_{tlt}^b\end{bmatrix} = T(\beta)\delta u_b\\
&U_{l,obq}^b = \sin(\psi^b)\left(U_{yw}^{\beta,gust} + \delta u_{yw}^b\right) - \cos(\psi^b)\left(U_{tlt}^{\beta,gust} + \delta u_{tlt}^b\right)\\
&U_n^{A,b} = U_n^{A,R} + \delta u_{ax}^b - \delta U_{ni}^{A,b} - \delta V_n^{A,b},\\
&U_l^{A,b} = U_l^{A,R} + U_{l,obq}^b + \delta V_l^{A,b},\\
&\alpha^{A,b} = \arctan\left(\tfrac{U_n^{A,b}}{U_l^{A,b}}\right) - \phi^{A,b}.
\end{aligned} \tag{5}$$

Step 6 *Compute normal and lead-wise forces and pitch-wise torques per blade element*

$$\begin{aligned}
\delta q_{f,n}^{A,b} &= \tfrac{1}{2}\rho c_A\left(C_L(\alpha^{A,b})U_l^{A,b} + C_D(\alpha^{A,b})U_n^{A,b}\right)\sqrt{\left(U_n^{A,b}\right)^2 + \left(U_l^{A,b}\right)^2} - \bar{q}_{f,n}^{A,b},\\
\delta q_{f,l}^{A,b} &= \tfrac{1}{2}\rho c_A\left(C_L(\alpha^{A,b})U_n^{A,b} - C_D(\alpha^{A,b})U_l^{A,b}\right)\sqrt{\left(U_n^{A,b}\right)^2 + \left(U_l^{A,b}\right)^2} - \bar{q}_{f,l}^{A,b},\\
\delta q_t^{A,b} &= -\tfrac{1}{2}\rho c_A^2 C_M(\alpha^{A,b})\left(\left(U_n^{A,b}\right)^2 + \left(U_l^{A,b}\right)^2\right) - \bar{q}_t^{A,b}.
\end{aligned} \tag{6}$$

Step 7 *Update dynamic term on axial induction speed δU_i^A to be used in next time instant (ECN Differential Equation Model):*

$$\begin{aligned}
&U_{tr}^{A,b} = \sqrt{\left(U_{ax}^{\beta,gust} + \delta u_{ax}^b - \bar{U}_i^A - \delta U_{ni}^{A,b}\right)^2 + \left(U_{yw}^{\beta,gust} + \delta u_{yw}^b\right)^2 + \left(U_{tlt}^{\beta,gust} + \delta u_{tlt}^b\right)^2},\\
&d^{A,b} = \tfrac{2\pi r_A}{3}\tfrac{U_{tr}^{A,b}}{\sqrt{\left(U_{tr}^{A,b}\right)^2 + \left(\bar{V}_l^{A,b} + \delta V_l^{A,b}\right)^2}},\\
&F_p^{A,b} = \tfrac{4}{\pi^2}\arccos\left(e^{-(R-r_A)\frac{\pi}{d^{A,b}}}\right)\arccos\left(e^{-(r_A-r_{root})\frac{\pi}{d^{A,b}}}\right),\\
&F_p^A = \tfrac{1}{3}\textstyle\sum_{i=1}^3 F_p^{A,b},\\
&U_{tr,F}^{A,b} = \sqrt{\left(U_{ax}^{\beta,gust} + \delta u_{ax}^b - F_p^A\bar{U}_i^A - \delta U_{ni}^{A,b}\right)^2 + \left(U_{yw}^{\beta,gust} + \delta u_{yw}^b\right)^2 + \left(U_{tlt}^{\beta,gust} + \delta u_{tlt}^b\right)^2},\\
&U_{tr,F}^A = \tfrac{1}{3}\textstyle\sum_{i=1}^3 U_{tr,F}^{A,b},\\
&f_a^A = 2\pi\Big/\textstyle\int_0^{2\pi}\tfrac{1-\frac{r_A}{R}\cos(\gamma)}{\left(1+\left(\frac{r_A}{R}\right)^2 - 2\frac{r_A}{R}\cos(\gamma)\right)^{\frac{3}{2}}}d\gamma,\\
&q_{n,lift}^A = \textstyle\sum_{b=1}^3\tfrac{1}{2}\rho c_A C_L(\alpha^{A,b})U_l^{A,b}\sqrt{\left(U_n^{A,b}\right)^2 + \left(U_l^{A,b}\right)^2},\\
&\tfrac{dU_i^A}{dt} = \tfrac{\frac{\delta r_A}{\rho S_A}q_{n,lift}^A - 2F_p^A U_i^A U_{tr,F}^A}{2RF_p^A f_a^A}\\
&\delta U_i^A \leftarrow \delta U_i^A + \tfrac{dU_i^A}{dt}T_s.
\end{aligned} \tag{7}$$

3.4 Conventional controller

The controller is built up of two loops: pitch control for generator speed regulation (active above rated only) and generator torque control for power regulation (according to optimal-λ QN-curve below rated, and constant power above rated). Both loops act on the rotor speed filtered with a series of low-pass filter at the 3P frequency (4th order inverse Chebyshev type II filter with cutoff frequency of $(3P - 0.8)$ rad/s and 20 dB reduction), band-stop filter around the first tower sidewards frequency f_{sd} (2nd order elliptic filter with stop-band $[0.85f_{sd}, 1.15f_{sd}]$ rad/s, 30 dB reduction and 1 dB ripple), and a band-stop filter at the first collective lead-lag frequency f_{ll} (4th order elliptic filter with stop-band $[0.8f_{ll}, 1.05f_{ll}]$ rad/s, 30 dB reduction and 1 dB ripple). The pitch controller is a PI compensator designed to achieve a gain margin of 2 and a phase margin of 45 degrees.

3.5 Problem Formulation

In this report, an extreme rising wind gust with simultaneous wind direction change is simulated. These have been chosen as specified in IEC 61400-1: 15 m/s rising wind gust (on top of the mean wind $\bar{U} = 15$ m/s and the additional blade-effective turbulence) in conjunction with a direction change of $720/\bar{U} = 48^o$. A simulation of the complete turbine model with the described extreme event occurring 5 sec after the beginning of the simulation, is shown in Figure 3. On the top subplot of the figure the rotor speed Ω_k (the fluctuating [black] curve), together with its filtered version Ω_k^f (the smoother [green] curve) are given. The rated speed $\bar{\Omega}$, being approximately 17.7 rpm is given by the bottom dotted line, while the overspeed limit, which should not be exceeded as this would trigger the supervisory system to start an emergency stop of the turbine, is given by the top dashed line. The overspeed limit is set to 15 % above the rated value (20.3 rpm).

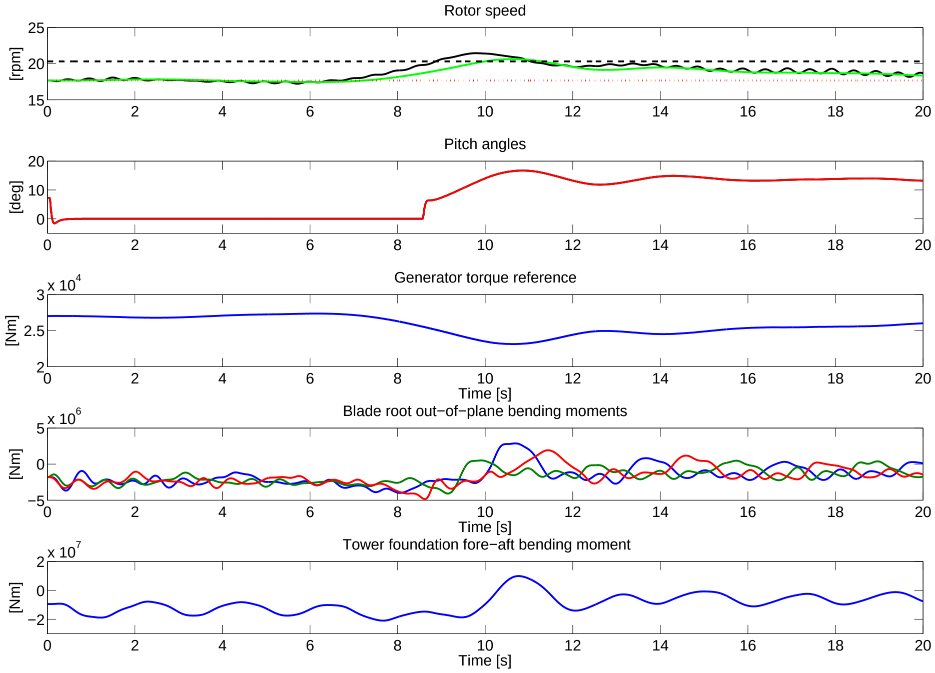


Figure 3: Turbine simulation under extreme rising gust and direction change at $t = 5$ sec, without EEC

The supervisory system is not modeled in the simulation, so the turbine is not stopped after the rotor speed exceeds the overspeed limit around $t = 9$ sec. The second subplot in Figure 3 gives the collective pitch angle of the rotor blades. In the beginning of the simulation the controller works at below-rated operation region, and switches to above rated when the filtered rotor speed exceeds 18.7 rpm ($= \bar{\Omega} + 1$ rpm). The third subplot (middle) shows the generator torque. The constant-power control strategy above rated is easily recognizable by the inverse proportionality of the generator torque to the filtered rotor speed. The fourth subplot gives the three flap-wise blade root bending moments. The 1p loads, resulting from the oblique inflow, are clearly seen in the second half of the simulation. Finally, the last (fifth) subplot in Figure 3 shows the tower base fore-aft bending moment.

The purpose of the report is to develop algorithm for extreme event control that

- is capable of preventing rotor overspeed, when possible, and
- achieves 1p blade root bending moment reduction.

To this end, the extreme event should be detected at an early stage, which is the focus of the next section.

4 Extreme Event Recognition

The recognition of extreme events, proposed here, is based on the estimation of the wind parameters u_b and β by means of a nonlinear estimator (EKF), which estimates are then used in a CUSUM test for detecting changes in their mean values as resulting from extreme wind gusts and/or extreme wind direction changes. This section describes these components in more detail.

4.1 Simplified model

The algorithm for EER utilizes an EKF for the estimation of a so-called *augmented state* x^a, consisting of the turbine structural model state x and the unknown inputs (i.e. the three blade effective wind speed signals u_b and the oblique inflow angle β). In order to somewhat reduce the computational complexity of the EKF, it is based on a more simplified model than the one used for turbine simulation, described in Section 3. This simplified model also consists of an interconnection of an SDM and ADM blocks, although their complexity is somewhat simplified as described below:

(ADM) The aerodynamics neglects the effects of the movement of the blades and tower onto the torques and forces acting on the blade elements (with the exception of the leadwise blade element velocity due to rotor rotation, which is, of course, not neglected). This boils down to setting $\delta V_l^{A,b} = \frac{\bar{V}_l^{A,b}}{\bar{\Omega}}(\Omega - \bar{\Omega})$ and $\delta V_n^{A,b} = 0$ in Section 3.3. Furthermore, the blade element pitch angle variations are assumed to be constant over the blade, i.e. $\delta\phi^{A,b} = \delta\phi^b$, and are assumed measured at the blade roots. The third simplification is that equilibrium wake is considered, being equivalent to setting $\delta U_i^A = 0$ (and skipping Step 7 in the algorithm of Section 3.3). The variations of the axial induction wind speed around the equilibrium value will then be (approximately) incorporated into the blade effective wind speed estimates as if there was equivalent longitudinal wind speed variation.

(SDM) The order of the structural model which is used for simulating the wind turbine (being 40), is reduced to 20 using the model reduction by balanced truncation technique. In this way, the 20 least controllable and observable states in the SDM model are removed. This model reduction is performed on the SDM model with all 130 inputs inputs, but only the 10 measured outputs (i.e. Ω, $\delta\phi^b$, M_x^{cm} and M_z^{cm}).

$(\boldsymbol{T_s})$ The model reduction, mentioned above, is performed after resampling the SDM model to $T_s^{ctr} = 0.02$ sec (the sampling time SDM for turbine simulation is $T_s = 0.005$ sec).

Define the Coleman transformation $T_M(\cdot)$ (modulation) and inverse Coleman transformation $T_D(\cdot)$ (demodulation)

$$T_D(\psi) \doteq \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 2\sin(\psi_1) & 2\sin(\psi_2) & 2\sin(\psi_3) \\ 2\cos(\psi_1) & 2\cos(\psi_2) & 2\cos(\psi_3) \end{bmatrix},\ T_M(\psi) \doteq \begin{bmatrix} 1 & \sin(\psi_1) & \cos(\psi_1) \\ 1 & \sin(\psi_2) & \cos(\psi_2) \\ 1 & \sin(\psi_3) & \cos(\psi_3) \end{bmatrix} = T_D^{-1}(\psi).$$

The map T_D is used to transform variables, defined in the rotating reference frame, to the non-rotating reference frame (e.g. $M_z^{cm} = T_D(\psi)M_z$), while T_M is used for the inverse operation. Using this notation, the simplified model can be compactly described in the following state-space form

$$
\begin{array}{l}
\textbf{Structural dynamics:}\\
x_{k+1} = Ax_k + B\delta v_k^{cm} + B_d \delta d_k^{cm}\\
\delta M_k^{cm} = Cx_k + D\delta v_k^{cm} + D_d \delta d_k^{cm}\\
\delta\Omega_k = C_\Omega x_k + D_\Omega \delta v_k^{cm}\\
\delta\phi_k^{cm} = C_\phi x_k + D_\Omega \delta v_k^{cm}\\
\textbf{Coleman (de)modulation:}\\
\delta M_k = (I_2 \otimes T_M(\psi_k))\delta M_k^{cm}\\
\delta\phi_k = T_M(\psi_k)\delta\phi_k^{cm}\\
\delta d_k^{cm} = (I_{3N_{ann}} \otimes T_D(\psi_k))\delta d_k\\
\delta v_k^{cm} = (T_D(\psi_k) \oplus 1)\delta v_k\\
\textbf{Aerodynamics:}\\
\delta d_k = f_{ADM}(\delta\Omega_k, \delta\phi_k, u_k, \beta_k)
\end{array}
\tag{8}
$$

where $x_k \in \mathbb{R}^n$ contains the (reduced) SDM model state,

$$\delta M_k^T = [\delta M_z^1, \delta M_z^2, \delta M_z^3, \delta M_x^1, \delta M_x^2, \delta M_x^3]_k,$$

is a vector of in-plane and out-of-plane blade root bending moments, $\delta v_k^T = [\delta\theta^T, \delta T_g]_k \in \mathbb{R}^4$ contains the control signals (being the reference blade pitch angles and generator torque), $u_k^T = [u_1, u_2, u_3]_k$ represents the blade-effective wind speeds, $\delta\phi_k^T = [\delta\phi^1, \delta\phi^2, \delta\phi^3]_k$ contains the blade pitch angles, and

$$\delta d_k = \text{vec}\left(\left[\begin{array}{ccc|ccc|ccc} \delta q_n^{1,1} & \dots & \delta q_n^{N_{ann},1} & \delta q_l^{1,1} & \dots & \delta q_l^{N_{ann},1} & \delta q_t^{1,1} & \dots & \delta q_t^{N_{ann},1}\\ \delta q_n^{1,2} & \dots & \delta q_n^{N_{ann},2} & \delta q_l^{1,2} & \dots & \delta q_l^{N_{ann},2} & \delta q_t^{1,2} & \dots & \delta q_t^{N_{ann},2}\\ \delta q_n^{1,3} & \dots & \delta q_n^{N_{ann},3} & \delta q_l^{1,2} & \dots & \delta q_l^{N_{ann},3} & \delta q_t^{1,3} & \dots & \delta q_t^{N_{ann},3} \end{array}\right]_k\right) \tag{9}$$

is a long vector consisting of all blade element normal and lead-wise force variations and pitch-wise torque variations. The function $f_{ADM}(\delta\Omega_k, \delta\phi_k, u_k, \beta_k)$ represents the equations (6), rewritten in terms of the variables $\{\Omega_k, \delta\phi_k, u_k, \beta_k\}$ by using equations (1)-(5) and under the simplifying assumptions for the ACM, described in the beginning of this section.

The following nonlinear model then relates the inputs to the measured outputs

$$
\begin{aligned}
x_{k+1} &= Ax_k + B(T_D(\psi_k) \oplus 1)\delta v_k + B_d(I \otimes T_D(\psi_k))f_{ADM}(\delta\Omega_k, \delta\phi_k, u_k, \beta_k)\\
\delta M_k &= (I \otimes T_M(\psi_k))\left(Cx_k + D(T_D(\psi_k) \oplus 1)\delta v_k + D_d(I \otimes T_D(\psi_k))f_{ADM}(\delta\Omega_k, \delta\phi_k, u_k, \beta_k)\right)\\
\delta\Omega_k &= C_\Omega x_k + D_\Omega(T_D(\psi_k) \oplus 1)\delta v_k,\\
\delta\phi_k &= T_M(\psi_k)(C_\phi x_k + D_\phi(T_D(\psi_k) \oplus 1)\delta v_k)
\end{aligned}
\tag{10}
$$

where the rotor azimuth ψ_k is viewed as known time-varying parameter since ψ_k is needed in $f_{ADM}(\delta\Omega_k, \delta\phi_k, u_k, \beta_k)$ but depends only on the rotor speed Ω up to time instant $(k-1)$, but not on Ω_k (and, hence, is not a function of the current state).

The goal is to construct a filter that uses the blade root bending moment measurements M_k to estimate the state x_k together with the unknown inputs u_k and β_k.

4.2 Augmented-state extended Kalman filter

For the purpose of EER, the unknown inputs u_k and β_k in model (10) need to be estimated. One way to do this is model them as the response of a given stochastic model to a random white noise process, to append this model to the turbine dynamics model and then use a Kalman filter to estimate both the state of the turbine and the state of the stochastic model from which u_k and β_k are computed. Although blade-effective wind turbulence models do exist (van Engelen and Schaak, 2007), their parametrization is in practice not an easy task. A much more practical approach is the so-called *augmented-state* Kalman filter technique, which is often used in the literature for the estimation of (time-varying) unknown input signals (disturbances), see e.g. Kanev and Verhaegen (2005) and the references therein. The basic idea behind this approach is to model the unknown input using a *random walk model*

$$\begin{bmatrix} u_{k+1} \\ \beta_{k+1} \end{bmatrix} = \begin{bmatrix} u_k \\ \beta_k \end{bmatrix} + r_k, \tag{11}$$

where r_k is a zero-mean white Gaussian process with covariance matrix R_r. Usually, the covariance matrix R_r of the noise term r_k is viewed as design parameter that provides a trade-off between tracking speed and smoothness of the estimates. For simplicity, it is often selected as diagonal matrix. Faster tracking of the true signals can be obtained by appropriately increasing the elements of R_r, which however results in less smooth (i.e. more noisy) estimates, and vice versa.

Basically, the model (11) represents an integrated white noise variable, so that the output will have its energy concentrated in the lower frequency band, and hence using such model is mostly suitable for modeling constant or slowly varying signals. The blade effective wind speeds and the wind orientation angle are naturally low frequency signals, making such kind of modeling sufficient. Given the random walk model (11), the state x of the system (10) is augmented with the unknown inputs, resulting in the following augmented-state model

$$\overbrace{\begin{bmatrix} x_{k+1} \\ u_{k+1} \\ \beta_{k+1} \end{bmatrix}}^{x^a_{k+1}} = \overbrace{\begin{bmatrix} Ax_k + B_d(I \otimes T_D(\psi_k)) f_{ADM}(\delta\Omega_k, \delta\phi_k, u_k, \beta_k) \\ u_k \\ \beta_k \end{bmatrix}}^{f(x^a_k, \psi_k)} + \overbrace{\begin{bmatrix} B(T_D(\psi_k) \oplus 1) \\ 0 \\ 0 \end{bmatrix}}^{\tilde{B}_k(\psi_k)} \delta v_k + \overbrace{\begin{bmatrix} 0 \\ I_4 \end{bmatrix}}^{E} r_k,$$

$$\begin{aligned} \delta M_k \quad = \quad & \overbrace{(I \otimes T_M(\psi_k)) \left(Cx_k + D_d(I \otimes T_D(\psi_k)) f_{ADM}(\delta\Omega_k, \delta\phi_k, u_k, \beta_k)\right)}^{g(x^a_k, \psi_k)} \\ & + \underbrace{(I \otimes T_M(\psi_k)) D(T_D(\psi_k) \oplus 1)}_{\tilde{D}_k(\psi_k)} \delta v_k, \end{aligned}$$

that, using the equations for $\delta\Omega_k$ and $\delta\phi_k$ in (10), can compactly be written in the form

$$\begin{aligned} x^a_{k+1} &= f(x^a_k, \psi_k) + \tilde{B}_k(\psi_k)\delta v_k + Er_k, \\ \delta M_k &= g(x^a_k, \psi_k) + \tilde{D}_k(\psi_k)\delta v_k + e_k. \end{aligned} \tag{12}$$

The signal e_k, which is included in (12), is a zero mean white Gaussian processes with covariance matrix R_e, that can be used to represent measurement noise. Of course, additional measurements can be added to the blade root bending moments in (12) such as the rotor speed and blade pitch setting angles, as in equation (10). However, this does not noticeably improve the quality of the estimation and hence the measurements $\delta\Omega_k$ and $\delta\phi_k$ will only be used to parameterize the nonlinear function $f_{ADM}(\delta\Omega_k, \delta\phi_k, u_k, \beta_k)$.

An extended Kalman filter (Boutayeb et al., 1997) can now be applied to the nonlinear state-space model (12) to estimate the augmented state x^a_k, containing the blade effective wind speeds u_k and the oblique inflow angle β_k. The EKF can be summarized as follows

Algorithm 4.1 (Extended Kalman Filter)

Initialization $\hat{x}_0^a = E\{x_0^a\}$, $P_0 = E\{(x_0^a - \hat{x}_0^a)(x_0^a - \hat{x}_0^a)^T\}$.

Step 1 *Compute*

$$A_{k-1} = \left.\frac{\partial f(x^a, \psi_k)}{\partial x^a}\right|_{x^a = \hat{x}_{k-1}^a}$$

Step 2 *Time update*

$$\begin{aligned} \hat{x}_{k|k-1}^a &= f(\hat{x}_{k-1}^a, \psi_k) + \tilde{B}_k \delta v_{k-1} \\ P_{k|k-1} &= A_{k-1} P_{k-1} A_{k-1}^T + E R_r E^T \end{aligned}$$

Step 3 *Compute*

$$C_k = \left.\frac{\partial g(x^a, \psi_k)}{\partial x^a}\right|_{x^a = \hat{x}_{k|k-1}^a}$$

Step 4 *Measurement update:*

$$\begin{aligned} K_k &= P_{k|k-1} C_k^T \left(C_k P_{k|k-1} C_k^T + R_e\right)^{-1} \\ \hat{x}_k^a &= \hat{x}_{k|k-1}^a + K_k(\delta M_k - g(x_{k|k-1}^a, \psi_k) - \tilde{D}_k \delta v_k) \\ P_k &= (I - K_k C_k) P_{k|k-1} \end{aligned}$$

Remark 4.1 *The EKF requires the partial derivatives of the nonlinear functions with respect to the state variables. These are analytically computed in Appendix A. Of course, they can also be computed numerically; however, this results in a significant increase of the computational burden, as well as in numerical inaccuracies. Another, still computationally involved, but derivative-free alternative to the EKF is the unscented Kalman filter (Wan and van der Merwe, 2000; Julier et al., 1995). The author's experience, however, is that for the model described here it often runs into numerical problems due to the output covariance matrix becoming numerically singular.*

4.3 CUSUM test for Extreme Event Detection

The EKF, discussed above, estimates the turbine structural model state x, together with the blade effective wind speed signals u and the oblique inflow angle β, contained in the augmented state x^a. Under normal conditions, u and β will be stochastic signals with zero mean value, while under extreme conditions their mean values will undergo a change. In order that appropriate extreme event control actions are triggered timely, it is necessary to be able to detect such mean value changes promptly (with small detection delay and no missed alarms), yet accurately (no false alarms). An algorithm that directly looks at the current values of the estimates $\hat{u}_k$ and $\hat{\beta}_k$ would be fast but too sensitive to noise and inaccuracies in the estimates, and would trigger many false alarms:

To circumvent this, a one-sided CUSUM test (Basseville and Nikiforov, 1993) is used here that offers a good speed/accuracy trade-off. This algorithm, in combination with the EKF, detects an extreme wind gust at a very early stage, before any significant increase of the (filtered) rotor speed. This makes it possible to react timely by pitching the blades, keeping the rotor speed within allowable limits. The algorithm can be summarized as follows

Algorithm 4.2 (CUSUM test)

Initialization *Choose integers k_u (moving window length), ν (insensitivity parameter), h (threshold) and set $\hat{u}_0^f = \hat{u}_0$ (vector with initial wind speed estimates), $\epsilon_0 = 0$.*

Update *Compute*

$$\begin{aligned}\hat{u}_k^f &= \frac{(k_u - 1)\hat{u}_{k-1}^f + \hat{u}_k}{k_u} \\ \epsilon_k &= \max\left(0, \epsilon_{k-1} + \hat{u}_k - \hat{u}_k^f - \nu\right).\end{aligned}$$

Detection *If* $(\|\epsilon_k\|_1 > h)$*, set* $f_{ee,k} = 1$*, else set* $f_{ee,k} = 0$*.*

The signal $\epsilon_k \in \mathbb{R}^3$, computed by the CUSUM test, remains small under normal circumstances. The first equation in the update step represents a moving average filter used to estimate the mean value of the three blade effective wind speed signals. If the wind speed estimate $\hat{u}_k$ starts increasing, ϵ_k will also increase until $\hat{u}_k$ converges, at which point $(\hat{u}_k - \hat{u}_k^f) < \nu$ and ϵ_k will start decreasing to zero again. In this way, an easy detection mechanism would be to put a threshold h on the sum of the elements of the vector ϵ_k, so that an extreme event flag is raised ($f_{ee,k} = 1$) whenever $\|\epsilon_k\|_1 > h$. Once $f_{ee,k}$ gets one, the EEC algorithm, described later on, will be activated, aiming at preventing rotor overspeed and reducing blade loads. This is the subject of the next section. It should be pointed out at this stage that the extreme event flag $f_{ee,k}$ can be pulled-down by either the CUSUM test algorithm above (i.e. when $\|\epsilon_k\|_1 \leq h$), or by the EEC algorithm itself (when it decides that no further pitching of the blades is necessary, see Algorithm 5.1). In the later case the extreme event might not have finished when the flag is pulled-down, but the EEC algorithm reckons no (further) action needed.

5 Extreme Event Control

This section develops and algorithm for EEC that consists of two parts:

- collective feedforward pitch control for preventing rotor overspeed,
- individual pitch control for blade load reduction.

These two control loops are described in more detail in the following subsections.

5.1 Rotor Overspeed Prevention

As already shown in the simulation on Figure 3, the conventional PI pitch controller is uncapable to keep the rotor speed within its limits under extreme wind gusts. The reason for that is that (a) it reacts on the filtered rotor speed Ω_k^f which is delayed by about 1 sec with respect to the true speed Ω_k, and (b) it does not respond quick enough. In order to react as fast as possible for preventing rotor overspeed, once an extreme event flag is raised by the CUSUM algorithm in Section 4.3, the EEC starts pitching the blades to feather with the maximally allowable pitch speed under extreme conditions $\dot{\theta}_{mx,ext}$. This results in fast reduction of the rotor speed, but has as a side effect a very large tower base fore-aft moment due to the large reduction of the rotor thrust force. In order to limit the tower base moment, after some time Δt_{eec} (about 1 sec) the pitching speed is reduced to the maximum pitch speed under normal conditions, $\dot{\theta}_{mx}$.

The conventional generator torque control at above-rated conditions was designed to achieve constant power, equal to the rated power (see Section 3.4). This implies a negative generator torque sensitivity to rotor speed variation, i.e. $\partial T_g / \partial \Omega < 0$. This has a destabilizing effect on the rotor speed, which is stabilized by the pitch control algorithm. However, due to the very slow dynamics of the pitch actuators, this results in higher oscillations of the rotor speed around its reference (rated) value. At extreme conditions, this destabilizing effect is removed by using a constant generator torque curve equal to the rated value $\bar{T}_g$. This results, of course, in an increase of the generated power of up to 10-15%. Whenever this is not acceptable for the power electronics, the original constant-power generator torque curve should be used.

The EEC for rotor overspeed prevention is switched off once the extreme event flag $f_{ee,k}$ is pulled down to zero by CUSUM algorithm in Section 4.3, or whenever the pitch angle θ_k gets “close” to a reference pitch angle $\theta_{ref,ext}(\hat{U}_{ax}^{\beta,gust})$, dependent on the estimated axial wind speed $\hat{U}_{ax,k}^{\beta,gust}$ (see equation (2))

$$\hat{U}_{ax,k}^{\beta,gust} = T_1(\hat{\beta}_k)\left(\bar{U} + \frac{1}{3}\sum_{b=1}^{3}\hat{u}_{b,k}\right), \tag{13}$$

where $T_1(\cdot)$ is defined in (1). More specifically, $\theta_{ref,ext}(\hat{U}_{ax,k}^{\beta,gust})$ is defined as the collective pitch angle that, for axial wind speed $\hat{U}_{ax,k}^{\beta,gust}$, rated rotor speed $\bar{\Omega}$ and rated generator torque $\bar{T}_g$, achieves azimuth-averaged static aerodynamic torque $\bar{T}_a = \bar{T}_g$. For a given $U_{ax}^{\beta,gust}$, $\theta_{ref,ext}$ is computed by solving the following nonlinear optimization problem

$$\theta_{ref,ext}(U_{ax}^{\beta,gust}) = \arg\min_{\theta} \|\bar{T}_a(\bar{\Omega}, \theta, U_{ax}^{\beta,gust}) - \bar{T}_g\|_2.$$

The function $\theta_{ref,ext}(U_{ax}^{\beta,gust})$ is numerically computed off-line and stored for different values of $U_{ax}^{\beta,gust}$. Simple linear interpolation is then performed on-line.

To avoid unnecessary on/off switchings of the EEC due to fluctuations in $\theta_{ref,ext}(\hat{U}_{ax}^{\beta,gust})$, hysteresis is introduced: the EEC will switch on only when the extreme event flag gets raised (i.e. $f_{ee,k} = 1$ and $f_{ee,k-1} = 0$) and the current collective pitch angle is at least $\Delta\theta_{ee}^{on}$ (e.g. 5^o) below the reference pitch angle. The extreme event flag gets pulled down to zero ($f_{ee,k} = 0$), implying EEC switch-off, by either the CUSUM test in Algorithm 4.2 (meaning that the extreme event has ended), or when the difference between the reference pitch angle $\theta_{ref,ext}(\hat{U}_{ax}^{\beta,gust})$ and the true current collective pitch angle drops below $\Delta\theta_{ee}^{off}$ (e.g. 4^o), meaning that no further EEC action is needed. The rotor speed limitation algorithm is can be summarized as follows.

Algorithm 5.1 (Collective EEC)

```
Initialization  Select $\Delta\theta_{ee}^{on}$, $\Delta\theta_{ee}^{off} < \Delta\theta_{ee}^{on}$, $t_{eec} = 0$.

Step 1  Use the current EKF estimates $\hat{u}_k$ and $\hat{\beta}_k$ to compute $\hat{U}_{ax,k}^{\beta,gust}$ using (13).

Step 2  Run CUSUM test in Algorithm 4.2. If $f_{ee,k} = 0$ then set $t_{eec} = 0$ and go to Step 5.

Step 3  Compute $\Delta\theta_{ee,k} = \theta_{ref,ext}(\hat{U}_{ax,k}^{\beta,gust}) - \frac{1}{3}\sum_{b=1}^{3}\phi_k^b$.

Step 4  If ($f_{ee,k-1} = 1$ and $\Delta\theta_{ee,k} \geq \Delta\theta_{ee}^{off}$) or ($f_{ee,k-1} = 0$ and $\Delta\theta_{ee,k} \geq \Delta\theta_{ee}^{on}$)
then
    switch conventional control off
    $t_{eec} \leftarrow t_{eec} + T_s^{ctr}$,
    $\theta_k = \begin{cases} \theta_{k-1} + \dot{\theta}_{mx,ext}T_s^{ctr} & \text{if } t_{eec} \leq \Delta t_{eec}, \\ \theta_{k-1} + \dot{\theta}_{mx}T_s^{ctr} & \text{otherwise}. \end{cases}$
    $T_{g,k} = \bar{T}_g$.
else
    $t_{eec} = 0$,
    $f_{ee,k} = 0$.

Step 5  If $f_{ee,k-1} = 1$ and $f_{ee,k} = 0$ then
    reinitialize conventional pitch control
    switch on conventional control.
```

Notice that the conventional pitch and generator torque controllers are switched off when the EEC becomes active. The selected EEC strategy causes no transient effects after the transition from conventional control to EEC. The inverse transition (back to conventional PI control), however, should be performed with much care since this can result in a very large transient. To prevent this, the conventional controllers are properly reinitialized before being switched on. This is described in Appendix B.

5.2 Blade load reduction

As mentioned in the beginning of Section 5, besides rotor overspeed prevention, an important issue under extreme wind gusts with direction change is the reduction of blade loads. A yawed wind inflow results in large 1p blade load variations (see Figure 3), and a 0p (i.e. static) rotor tilt moment, that can be reduced by means of individual blade pitch control. This is the purpose of this section.

For IPC control design purposes, the nonlinear model (8) is linearized at a given operating point, resulting in the following linear model in Coleman domain

$$\mathcal{T}: \left\{ \begin{array}{rcl} x_{k+1} & = & \tilde{A}x_k + \tilde{B}\theta_k^{ty} + \tilde{B}_u u_k^{ty}, \\ M_k^{ty} & = & \tilde{C}x_k + \tilde{D}\theta_k^{ty} + \tilde{D}_u u_k^{ty} \end{array} \right.$$

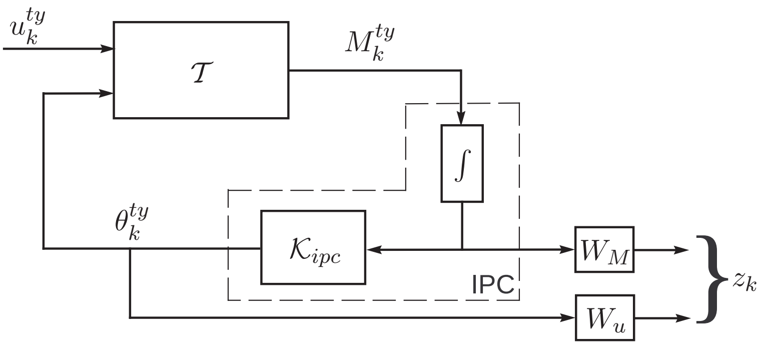


Figure 4: Block scheme for IPC design.

where the signals u_k^{ty}, θ_k^{ty} and M_k^{ty} contain the tilt and yaw oriented components of the *multi-blade* blade effective wind speeds u_k^{cm}, blade pitch angles θ_k^{cm} and flapwise blade root bending moments M_z^{cm}, respectively[1]. The considered extreme event in this report (gust with direction change) can be modeled by a nonzero constant tilt-oriented (i.e. first) component in u_k^{ty}. The collective pitch control loop has only a negligible influence on the rotor tilt and yaw moments and has been left out for simplicity. Similarly, the controls θ_k^{ty} also barely affect the rotor speed dynamics and need not be taken into consideration in the conventional rotor speed control design.

The goal here is to design a stabilizing controller that uses the rotor moments M_k^{ty} as inputs and computes the control actions θ_k^{ty} so as to minimize the low frequency components of the rotor moments' signals. In the rotating reference frame this corresponds to the suppression of 1p load components in the blades. In order to achieve zero steady state rotor moments, an integral action will be included in the controller. Furthermore, the control action should not be too active at certain frequencies, excited by the external wind disturbance, such as the 3p frequency f_{3P}, and eventually the 6p frequency f_{6P} and the first tower frequency f_{tow}. In addition to that, no high frequency control activity is desired.

To achieve all these performance specifications, an $\mathcal{H}_\infty$-optimal controller with integral action will be designed, optimizing the transfer from the external inputs u_k^{ty} to some suitable chosen weighted versions of the rotor moments and control action. More specifically, Figure 4 provides an block-schematic view of the IPC design model. In order to include integral action into the controller, the output of the system $\mathcal{T}$ is appended with integrators (one integrator per output), which integrated model is used for an optimal $\mathcal{H}_\infty$ controller design $\mathcal{K}_{ipc}$. Once designed, the final controller is constructed by moving the integrators, used in the design model, to the inputs of the computed controller (see the area inside the dashed curve on Figure 4).

Of course, an optimal controller designed based on the linearized turbine model $\mathcal{T}$ will only remain optimal at the working point at which the model is linearized. As the working point continually changes, it is important that once the controller has been designed, its stability and performance are evaluated at different working points. To achieve improved robustness properties to unmodelled dynamics, an $\mathcal{H}_\infty$ controller is designed. It should be pointed out that it is relatively simple to achieve better performance throughout the whole operation range of the turbine by means of gain-scheduling. To this end, an approach similar to the conventional way of including gain-scheduling collective pitch control algorithms (van der Hooft et al., 2003) can be

[1] Note that the tilt and yaw components (u_k^{ty}) of the *multi-blade* wind signals should not be mistaken with the tilt and yaw oriented components of the wind velocity vector relative to the rotor plane (see Figure 1). The former are obtained as a result of the Coleman transformation of the three axial blade effective wind speeds and are such that the yaw-oriented (tilt-oriented) component of u_k^{ty} affects (mainly) the yaw (tilt) rotor moment. On the other hand, the yaw-oriented (tilt-oriented) component of the wind velocity vector mainly affects the tilt (yaw) rotor moment, respectively.

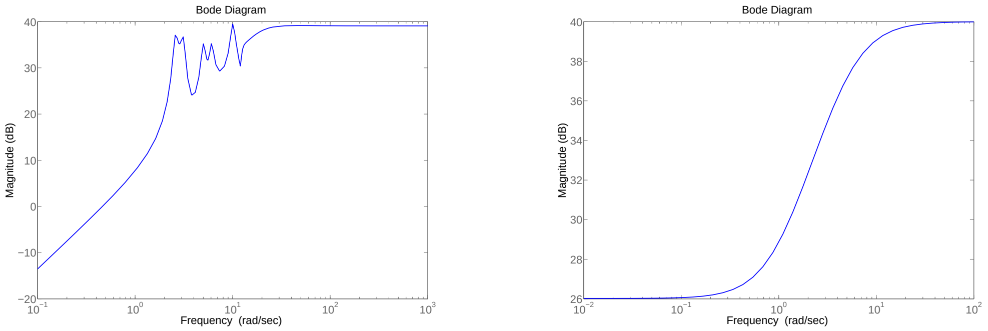


Figure 5: Bode magnitude plots of the weighting functions W_u (left) and W_M (right)

used, i.e. the gain of the IPC controller can be scheduled as a function of the pitch angle in such a way that the DC gain of the resulting open-loop transfer function remains constant. Although this approach falls outside the scope of this report, in a practical application gain-scheduling of the IPC controller needs to be considered.

In order to comply with these frequency domain design specifications, the controller $\mathcal{K}_{ipc}$ is designed by minimizing the $\mathcal{H}_\infty$ norm of the closed-loop transfer from the external inputs u_k^{ty} to the weighted integrated rotor moments and weighted control signals, as shown in Figure 4 (see the generalized output signal z_k). To this end, two weighting functions, W_M and W_u, can be selected with Bode magnitude plots as shown on Figure 5. For producing the left subplot on Figure 5, the weighting function for the control signals has been chosen as

$$W_u(z) = 10(F_{hp}(z) + F_{3p}(z)F_{6p}(z)F_{tow}(z) - 2)I_2, \tag{14}$$

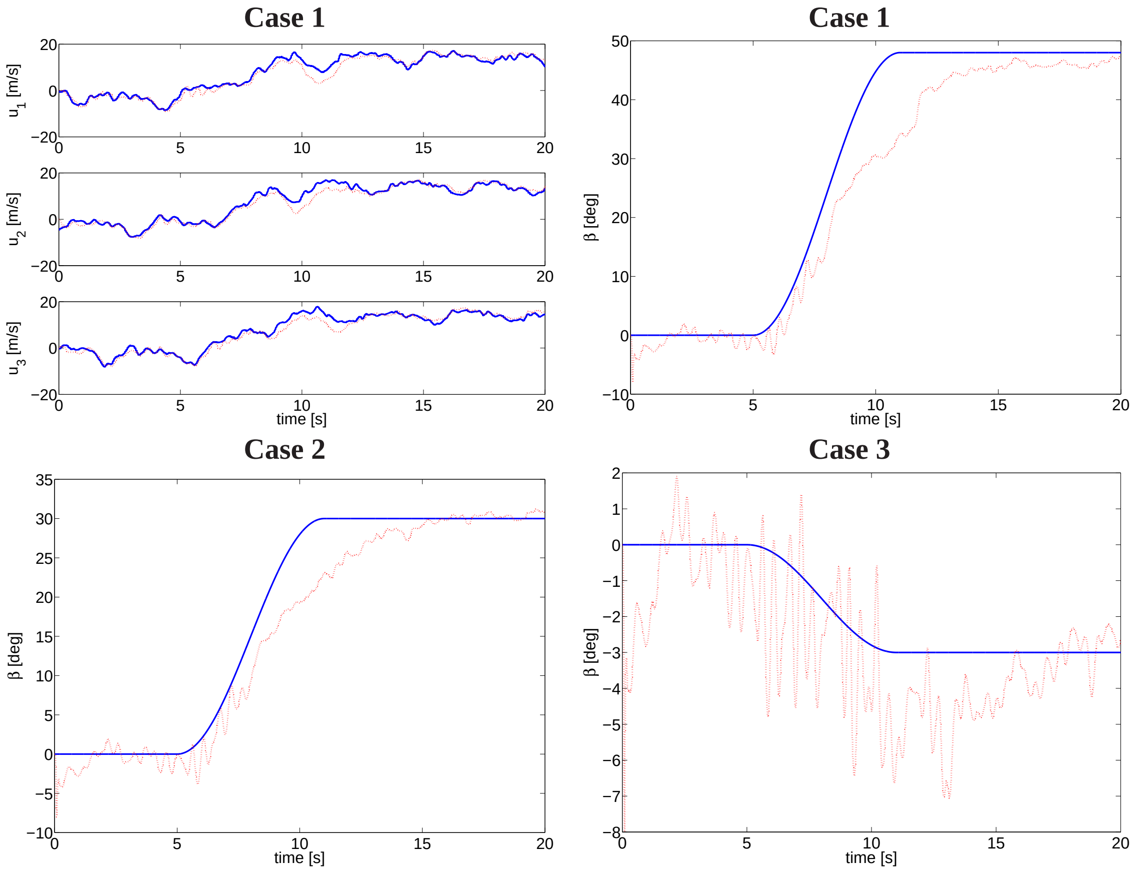
where $F_{hp}(z)$ is a second order inverse Chebyshev high-pass filters (frequency $f_{hp} = 4P$, reduction 20 dB, ripple 1dB), and $F_{3p}(z)$, $F_{6p}(z)$ and $F_{tow}(z)$ are second order inverse Chebyshev bandpass filters with the same reduction and ripple and bandpass intervals of $[0.9, 1.1]f_{3P}$, $[0.9, 1.1]f_{6P}$ and $[0.9, 1.1]f_{tow}$, respectively. All filters have been scaled to achieve unity DC-gain, so that W_u computed via (14) has a DC gain of zero. The so-selected weighting function W_u punishes control activity at frequencies f_{tow}, f_{3P}, f_{6P} and higher. The weighting function W_M, on the other hand, puts a frequency domain weighting on the integrated rotor moments. As there is integral action in the controller anyway, the lower frequencies need not to be weighted additionally. Instead, W_M could be used to eventually put some weighting on certain frequencies within the desired controller bandwidth which are otherwise not sufficiently actuated by the integral type control action. The weighting function W_M used for producing the right subplot in Figure 5 is a lead-lag filter with lead frequency of 1 rad/sec, lag frequency of 5 rad/sec and DC-gain of 20. Notice that W_M acts on the integrated rotor moments. Translating this to the original the rotor moments M^{ty}, this results in some additional weighting of the frequency band $[1, 5]$ rad/sec.

The augmented plant with the integrators and the weighting filters has then the following transfer function

$$\mathcal{T}^a(z) : \left\{ \; z_k = \begin{bmatrix} \begin{bmatrix} 0 & W_u(q^{-1}) \end{bmatrix} \\ \frac{T_s^{ctr}}{1-q^{-1}} W_M(q^{-1}) \mathcal{T}(q^{-1}) \\ \mathcal{T}(q^{-1}) \end{bmatrix} \begin{bmatrix} u_k^{ty} \\ \theta_k^{ty} \end{bmatrix} . \right.$$

The $\mathcal{H}_\infty$ optimal controller for $\mathcal{T}^a(z)$ is computed via the following optimization problem

$$\mathcal{K}_{ipc} = \arg\min_K \|\mathcal{F}(\mathcal{T}^a(z), K(z))\|_\infty,$$



Case 1

Case 1

Case 2

Case 3

Figure 6: Simulated (solid blue) and estimated (dotted red) blade effective wind speeds u_b (top, left) for case 1, and oblique inflow angle β (right) for case 1 (top, right), case 2 (bottom, left) and case 3 (bottom, right)

where $\mathcal{F}(\mathcal{T}^a(z), K(z))$ denotes the closed-loop system, $\| \cdot \|_\infty$ denote the $\mathcal{H}_\infty$ system norm, and wherein the optimization is defined over all controllers $K(z)$ that have the same number of states as the augmented model $\mathcal{T}^a(z)$. For more details on modern robust control design, the reader is referred to Zhou and Doyle (1998). The controller $\mathcal{K}_{ipc}$, designed in this way, will be a MIMO (2-by-2) transfer function, mapping the *integrated* rotor tilt and yaw moments to the tilt and yaw oriented blade pitch angles. Moving the integrators back to the controller results in the final IPC

$$\mathcal{K}_{ipc}^i = \mathcal{K}_{ipc} \begin{bmatrix} \frac{T_s^{ctr}}{z-1} & \\ & \frac{T_s^{ctr}}{z-1} \end{bmatrix} .$$

6 Simulation

The performance of the complete algorithm, including extreme event recognition and control, is demonstrated on simulation data, obtained with the nonlinear test turbine model described in Section 3. The model represents a 3-bladed HAWT with rated power of 2.5MW, rotor radius of $R = 40$ m, and rated rotor speed of $\bar{\Omega} = 1.85$ rad/sec. In the BEM module, the blades are represented by $N_{ann} = 15$ elements. The structural model is linearized around an equilibrium point corresponding to rated rotor speed, mean longitudinal wind speed of $\bar{U} = 15$ m/s (with $\bar{\phi}_{tlt} = -5.138^o$ [mainly due to tilted rotor] and $\bar{\phi}_{yw} = 0.01^o$) and blade pitch angles of $\bar{\phi}^b = 7.24^o$. The values selected for the tuning parameters of the EER and EEC schemes are given in Table 2.

In order to evaluate the performance of the proposed algorithm under different wind gust conditions, three different cases are simulated, as summarized in Table 3. The first case corresponds to the *extreme direction change* (EDC) as specified in the norm IEC 61400-1. The EDC consists of a rising $V_{gust} = 15$ m/s wind gust with a simultaneous wind direction change of $\beta_{gust} = 720/\bar{U}$ degrees. The effects of this on the turbine loads have been described in Section 3.5. The second case corresponds to the same rising wind gust ($V_{gust} = 15$ m/s) but a different, smaller wind direction change angle ($\beta_{gust} = 30$ degrees). This results in even larger 1p loads on the blades as compared to the first case due to the much larger axial component of the wind velocity vector, i.e. $\cos(\beta_{gust})(\bar{U} + V_{gust})$. Hence, the second case has the purpose to test the capabilities of the proposed algorithm to even more serious wind gust conditions, than specified in the IEC norm. The third case, on the other hand, has the purpose to test whether the algorithm is not overly sensitive, and is not responding to minor events, which is not desirable as the conventional controller should be able to handle them. For that purpose, the third case comprises a 3 m/s wind gust in combination with a -3 degrees direction change. This last case should not trigger the EEC algorithm.

Different simulations are run, each being 50 sec long. The turbine dynamics is simulated at a sample rate of 200 Hz, while the controllers (CPC and IPC) work at 50 Hz. In the time series,

Alg.	Variable	Value	Description
EKF	n	20	state dimension
	$\hat{x}_0^a$	0	initial state estimate
	P_0	$\begin{bmatrix} 10^{-4} I_{n+3} & \\ & 10^{-5} \end{bmatrix}$	initial state covariance matrix
	R_r	$\begin{bmatrix} 10^{-2} I_2 & \\ & 10^{-4} \end{bmatrix}$	process noise covariance matrix
	R_e	$\begin{bmatrix} 10^{3} I_3 & \\ & 10^{2} I_3 \end{bmatrix}$	measurement noise covariance matrix
Cusum	k_u	25	moving window length
	ν	1	insensitivity parameter
	h	100	threshold
EEC	$\dot{\theta}_{mx,ext}$	$10^o/s$	max pitch speed under extreme event
	$\dot{\theta}_{mx}$	$4^o/s$	max pitch speed under normal conditions
	$\Delta\theta_{ee}^{on}$	5^o	EEC activation zone
	$\Delta\theta_{ee}^{off}$	4^o	EEC deactivation zone
	$F_{3p}(z)$	$\frac{z^4-3.973z^3+5.948z^2-3.977z+1.002}{z^4-3.953z^3+5.883z^2-3.908z+0.9774}$	3p band pass filter
	$F_{6p}(z)$	1	6p band pass filter
	$F_{tow}(z)$	1	tower frequency band pass filter
	$F_{hp}(z)$	$\frac{10z^2-19.48z+9.57}{z^2-1.554z+0.6415}$	control signal weighting filter
	$F_M(z)$	$\frac{100z-98.1}{z-0.9048}$	integrated rotor moments lead-lag filter

Table 2: Parameters used in the described algorithms.

case	1	2	3
V_{gust} [m/s]	15	15	3
β_{gust} [deg]	48	30	-3

Table 3: Simulated wind gust cases

presented in the figures below, only the first 20 seconds are plotted. The (extreme) events occur 5 sec from the beginning of each simulation. For the power spectra plots later on, the time series from the 10th sec to the end of the simulations are used, so that only the data after the event occurrence (and after the transients have died out) is taken. The first two cases are simulated two times, once with the EEC algorithm turned off (i.e. conventional controller active all the time), and once with the EEC algorithm turned on. This makes it possible to investigate to what extend the proposed EEC algorithm improves on the rotor speed control and load reduction under extreme gust conditions. The third case is simulated only once, since even when the EEC algorithm is turned on, it does no get activated by the EER scheme as the event is not recognized as major.

Evaluation of the EER

The performance of the EER scheme is determined by the accuracy of the estimates of the EKF. To evaluate that, we will compare the simulated blade effective wind speeds u_b and the simulated wind direction change angle β to their estimates, computed by the EKF. Other quantities have also been investigated, though not reported here as these are not used in the change detection mechanism.

Figure 6 shows the performance of the EKF scheme under the three simulated scenarios. The top left subplot represents the three simulated blade effective wind speeds (solid blue curves) and their estimates (dotted red curves) by the EKF for case 1 only. The excellent accuracy of the wind estimates remains unchanged under cases 2 and 3, though these are not reported here for the sake of brevity. The remaining three subplots in Figure 6 depict the simulated oblique inflow angle β (solid blue curves) together with its EKF estimates (dotted red curves) for the three different cases. Clearly, these estimates are sufficiently accurate for the detection of wind direction changes wince the estimates do not differ more than about ± 3 degrees from the simulated values.

Evaluation of the EEC

As discussed in Section 3.5, the purpose of the EEC algorithm is to prevent rotor overspeed (that can trigger unnecessary emergency shutdown of the turbine) and to reduce large blade 1p loads under extreme wind gust conditions. On the other hand, the EEC algorithm should remain inactive under mild gust conditions. To demonstrate its performance, the rotor speed Ω, the blade pitch angles ϕ^b and the blade root out-of-plane bending moments M_z^b are next investigated under the above-mentioned three load cases. Figure 7 pertains to load case 1, where the subplots on the left hand side correspond to the case without EEC, while the subplots on the right – to the case with EEC. Clearly, when the EEC algorithm is not present, this load case leads to the rotor speed Ω getting much above its limit. This is due to the conventional controller remaining in partial load regime until the *filtered* rotor speed Ω^f (dashed green line) exceeds the rated speed $\bar{\Omega}$ by 1rpm, at which point the true speed Ω is already too large. The EEC algorithm, on the other hand, detects the gust at an early stage (at time 6.125 seconds) and starts pitching the blades to feathering position, preventing rotor overspeed (see top and middle right-hand side subplots). Moreover, once the estimated oblique inflow angle exceeds 10 degrees (the red dashed curve on top-right subplot in Figure 6), the IPC control is activated achieving substantial blade load reduction, as observed by comparing the bottom subplots on Figure 7 during the second half of the simulation (where the IPC is active). The achieved blade load reduction can be also appreciated by observing the left subplot on Figure 9 that depicts the spectra of the blade root out-of-plane bending moment variations δM_z^b in the cases without (solid red curve) and with (dashed black curve) EEC. The

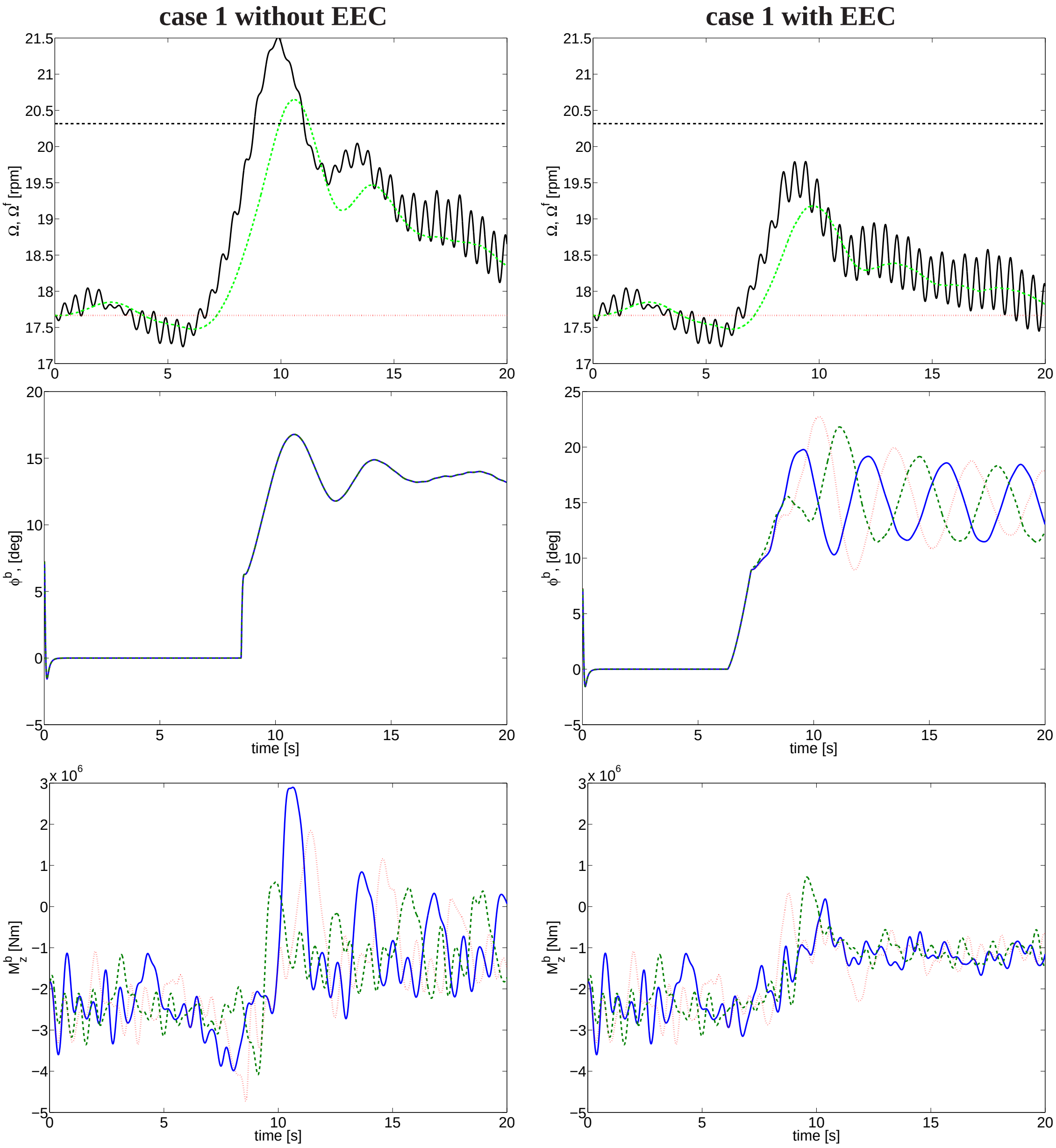


Figure 7: Turbine simulation under case 1 (extreme 15 m/s rising gust and 48 deg direction change at $t = 5$ sec) without EEC (left) and with EEC (right)

simulation results under case 2 are depicted in Figure 8. Again, the subplots on the left hand side correspond to the case without EEC, while the subplots on the right – to the case with EEC. As already mentioned, this load case is even more serious than the first one. This can indeed be seen by observing that the rotor speed (top left subplot in Figure 8) rises to as much as 23 rpm (i.e. more than 30% above the rated value). Similarly, the 1p blade loads also have a much higher amplitude as compared to case 1. With EEC, again, the rotor speed remains within its limits (top right subplot in Figure 8), while the IPC action, initiated after the oblique inflow angle exceeds 10 degrees, achieves significant 1p blade load damping, as can be seen from the bottom right subplot in Figure 8, as well as from the power spectra in the right-hand side subplot of Figure 9.

Finally, case 3 is simulated only once, i.e. with the EEC algorithm on, although it does not get activated by the EER scheme since the simulated event does not get recognized as a major one by the CUSUM test. As a result, the conventional controller remains active through the whole

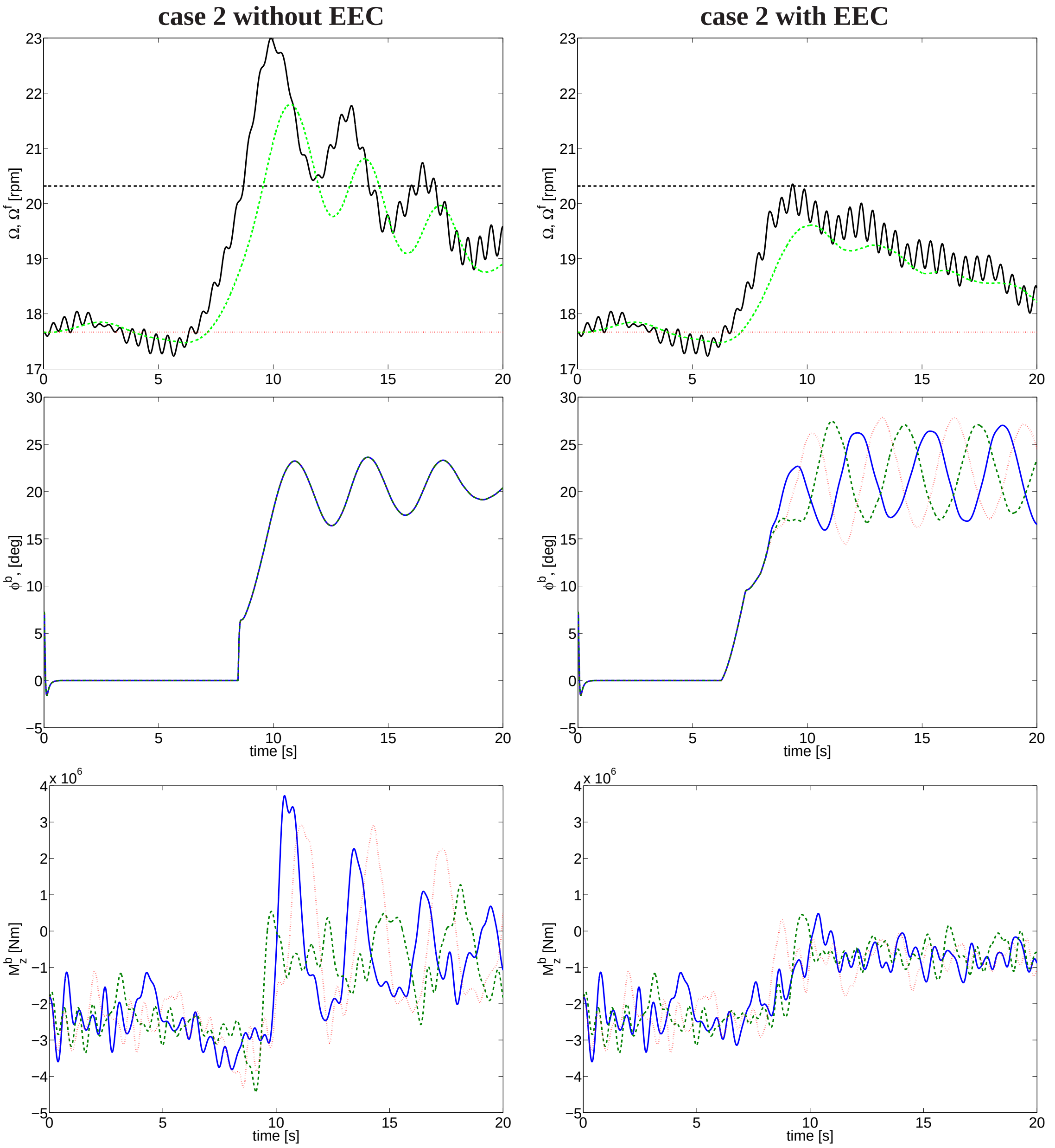


Figure 8: Turbine simulation under case 2 (extreme 15 m/s rising gust and 30 deg direction change at $t = 5$ sec) without EEC (left) and with EEC (right)

simulation. The rotor speed Ω, the blade pitch angles ϕ^b, and the blade root out-of-plane bending moments M_z^b are given in Figure 10. It can be observed, indeed, that no EEC is necessary in this case as the rotor speed remains well within its limits, and the blade root bending moments M_z^b after the event occurrence remain comparable to those before the gust.

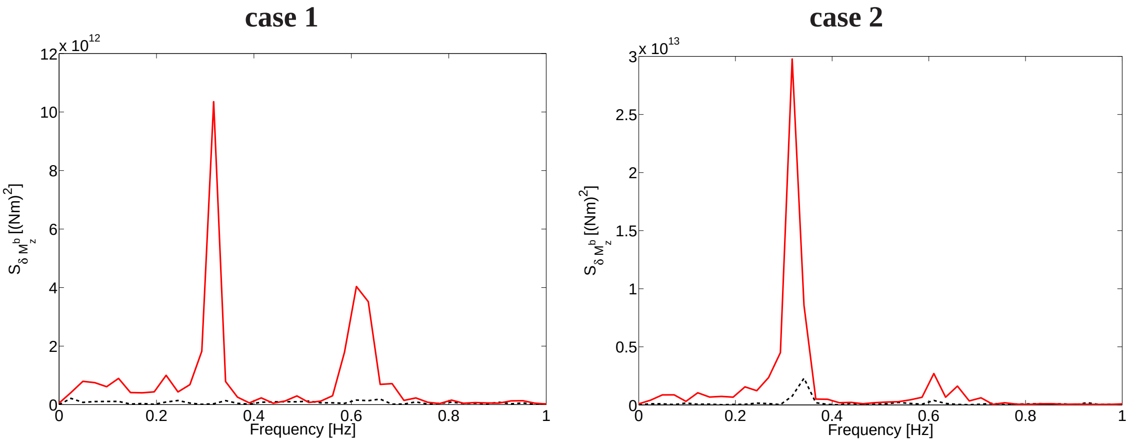


Figure 9: PSD of blade root flapwise bending moments M_z^b for case 1 (left) and case 2 (right), without EEC (solid curves) and with EEC (dashed curves)

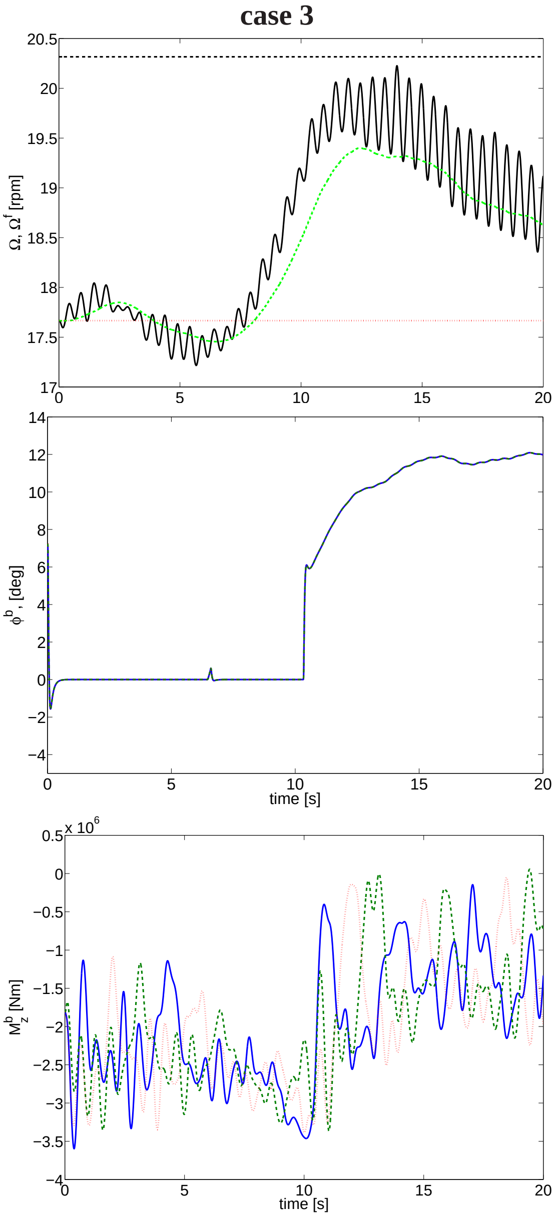


Figure 10: Turbine simulation under case 3 (3 m/s rising gust and -3 deg direction change at $t = 5$ sec). Due to the mild gust condition, the EEC does not get activated.

7 Conclusion

Extreme wind gust with direction change can cause turbine shutdown due to rotor overspeed, and can lead to a significant increase of blade 1p loads. The conventional pitch control algorithm, acting on the filtered rotor speed, reacts to the wind gust with a large delay caused by the large rotor inertia and the delay introduced by the rotor speed filter. This delay, combined with the intrinsically calm reaction of the conventional PI regulator, can easily lead to rotor overspeed, as demonstrated in this report. To avoid this, an algorithm for extreme event recognition and control is developed that uses (a) an EKF to estimate the turbine states together with the blade-effective wind speeds and oblique wind inflow angle, (b) a CUSUM algorithm to detect changes in the mean of the estimated wind signals, (c) a fast feedforward collective pitch control algorithm for rotor overspeed prevention, and (d) a feedback individual pitch control algorithm for 1p blade load reduction. The complete algorithm is demonstrated in different nonlinear simulations with a 40th order (linearized) structural dynamics model, obtained with the software TURBU, detailed nonlinear BEM aerodynamics and realistic blade-effective wind speed signals.

References

Basseville, M. and V. Nikiforov (1993): *Detection of Abrupt Changes - Theory and Application*. Prentice-Hall, Englewood Cliffs, N.J. URL `http://www.irisa.fr/sisthem/kniga/`.

Boutayeb, M., R. Rafaralahy and M. Darouach (1997): *Convergence Analysis of the Extended Kalman Filter Used as an Observer for Nonlinear Deterministic Discrete-Time Systems*. IEEE Transactions on Automatic Control, 42(4):581–586.

Goodwin, G., F. Graebe and M. Salgado (2001): *Control System Design*. Prentice Hall, Upper Saddle River, New Jersey.

Julier, S., J. Uhlmann and H. Durrant-Whyte (1995): *A new approach for filtering nonlinear systems*. Proceedings of the American Control Conference. pages 1628–1632.

Kanev, S. and M. Verhaegen (2005): *Two-Stage Kalman Filtering via Structured Square-Root*. Journal of Communications in Informations and Systems, 5(1):143–168.

Kodama, N. and T. Matsuzaka (2000): *Power Variation Control of a Wind Turbine Generator using Probabilistic Optimal Control, including Feed-Forward Control from Wind Speed*. Wind Engineering, 24(1):13–23.

Lindenburg, C. and G. Schepers (2001): *PHATAS-IV Aeroelastic modelling: Release "DEC-1999" and "NOV-2000"*. Report ECN-CX–00-027, Energy Research Center of the Netherlands.

Ma, X., M. Poulsen and H. Bindner (1995): *Estimation of Wind Speed in Connection to a Wind Turbine*. Report IMM-Technical report-1995-26, Technical University of Denmark.

Østergaard, K., P. Brath and J. Stoustrup (2007): *Estimation of effective wind speed*. Proceedings of the conference on the science of making torque from wind.

Rodriguez, J., J. Romagnoli and G. Goodwin (2003): *Supervisory multiple regime control*. Journal of Process Control, 177-191:13(2).

Sbarbaro, D. and R. Peña (2000): *A non-linear wind velocity observer for a small wind energy system*. Proceedings of the 39th IEEE Conference on Decision and Control.

Snel, H. and G. Schepers (1994): *Joint Investigation of Dynamic Inflow Effects and Implementation of an Engineering Method*. Report ECN-C–94-107, Energy Research Center of the Netherlands (ECN).

Thiringer, T. and A. Petersson (2005): *Control of a Variable-Speed Pitch-Regulated Wind Turbine*. Report, Chalmers University of Technology.

Turner, M. and D. Walker (2000): *Linear Quadratic Bumpless Transfer*. Automatica, 36(8):1089–1101.

van der Hooft, E., P. Schaak and T. van Engelen (2003): *Wind turbine control algorithms*. Report ECN-C–03-111, Energy Research Center of the Netherlands (ECN). URL `http://www.ecn.nl/publicaties/default.aspx?nr=ECN-C--03-111`. DOWEC-F1W1-EH-03-094/0.

van der Hooft, E. and T. van Engelen (2004): *Estimated Wind Speed Feedforward Control for Wind Turbine Operation Optimization*. Report ECN-C–04-126, Energy Research Center of the Netherlands (ECN).

van der Hooft, E., T. van Engelen, J. Pierik and P. Schaak (2007): *Real-time Process Simulator for Evaluation of Wind Turbine Systems: Modelling and Implementation*. Report ECN-E–07-046, Energy Research Center of the Netherlands (ECN).

van Engelen, T. (2007): *Control design based on aero-hydro-servo-elastic linear models from TURBU (ECN)*. Proceeding of the European Wind Energy Conference. Milan, Italy.

van Engelen, T. and P. Schaak (2007): *Oblique Inflow Model for Assessing Wind Turbine Controllers*. Proceedings of the 2nd Conference on The Science of Making Torque From Wind. Denmark.

Wan, E. and R. van der Merwe (2000): *The Unscented Kalman Filter for Nonlinear Estimation*. Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium. Lake Louise, Alberta, Canada.

Zhou, K. and J. Doyle (1998): *Essentials of Robust Control*. Prentice-Hall.

A Computing $\partial f/\partial x$ and $\partial g/\partial x$ in the EKF

In the extended Kalman filter, summarized in Section 4.2, the derivatives of the nonlinear terms $f(\cdot)$ and $g(\cdot)$ are needed with respect to the elements of the augmented state vector x^a. This subsection derives analytical expressions for these derivatives.

Since $\delta\Omega_k$ and $\delta\phi_k$ are functions of the state x_k and the input δv_k, $f_{ADM}(\delta\Omega_k, \delta\phi_k, u_k, \beta_k)$ will be written as $f_{ADM}(x_k, \delta v_k, u_k, \beta_k)$, wherein additionally the time index will be dropped for notational simplicity. Then, because $\partial\psi_k/\partial\delta\Omega_k = 0$, it holds that

$$\begin{aligned}
\tfrac{\partial f(x^a,\psi_k)}{\partial x^a} &= \begin{bmatrix} A & & \\ & 0 & \\ & & 0 \end{bmatrix} + \begin{bmatrix} B_d \\ 0 \\ 0 \end{bmatrix} (I \otimes T_D(\psi_k)) \begin{bmatrix} \tfrac{\partial}{\partial x^T} & \tfrac{\partial}{\partial u^T} & \tfrac{\partial}{\partial \beta} \end{bmatrix} f_{ADM}(x, \delta v, u, \beta) \\
\tfrac{\partial g(x^a,\psi_k)}{\partial x^a} &= (I \otimes T_M(\psi_k)) \left(\begin{bmatrix} C & 0 & 0 \end{bmatrix} + D_d (I \otimes T_D(\psi_k)) \begin{bmatrix} \tfrac{\partial}{\partial x^T} & \tfrac{\partial}{\partial u^T} & \tfrac{\partial}{\partial \beta} \end{bmatrix} f_{ADM}(x, \delta v, u, \beta) \right)
\end{aligned}$$

From (8)-(9) we see that $f_{ADM}(\cdot)$ is a vector containing the blade element torques $\delta q_t^{A,b}$, normal $\delta q_{f,n}^{A,b}$ and leadwise $\delta q_{f,l}^{A,b}$ forces, the derivatives of which with respect to $x \in \mathbb{R}^n$, $u \in \mathbb{R}^3$ and β need to be found.

Denote the i-th element of the vector x^a as x_i^a. Then from (6) follows

$$\begin{aligned}
\frac{\partial \delta q_{f,n}^{A,b}}{\partial x_i^a} &= \frac{1}{2}\rho c_A \left(\frac{\partial C_L(\alpha^{A,b})}{\partial \alpha^{A,b}} \frac{\partial \alpha^{A,b}}{\partial x_i^a} U_l^{A,b} + \frac{\partial C_D(\alpha^{A,b})}{\partial \alpha^{A,b}} \frac{\partial \alpha^{A,b}}{\partial x_i^a} U_n^{A,b} \right) \sqrt{\left(U_n^{A,b}\right)^2 + \left(U_l^{A,b}\right)^2} \\
&+ \frac{1}{2}\rho c_A \left(C_L(\alpha^{A,b}) \frac{\partial U_l^{A,b}}{\partial x_i^a} + C_D(\alpha^{A,b}) \frac{\partial U_n^{A,b}}{\partial x_i^a} \right) \sqrt{\left(U_n^{A,b}\right)^2 + \left(U_l^{A,b}\right)^2} \\
&+ \frac{1}{2}\rho c_A \left(C_L(\alpha^{A,b}) U_l^{A,b} + C_D(\alpha^{A,b}) U_n^{A,b} \right) \frac{U_n^{A,b} \frac{\partial U_n^{A,b}}{\partial x_i^a} + U_l^{A,b} \frac{\partial U_l^{A,b}}{\partial x_i^a}}{\sqrt{\left(U_n^{A,b}\right)^2 + \left(U_l^{A,b}\right)^2}}.
\end{aligned} \tag{15}$$

$$\begin{aligned}
\frac{\partial \delta q_{f,l}^{A,b}}{\partial x_i^a} &= \frac{1}{2}\rho c_A \left(\frac{\partial C_L(\alpha^{A,b})}{\partial \alpha^{A,b}} \frac{\partial \alpha^{A,b}}{\partial x_i^a} U_n^{A,b} - \frac{\partial C_D(\alpha^{A,b})}{\partial \alpha^{A,b}} \frac{\partial \alpha^{A,b}}{\partial x_i^a} U_l^{A,b} \right) \sqrt{\left(U_n^{A,b}\right)^2 + \left(U_l^{A,b}\right)^2} \\
&+ \frac{1}{2}\rho c_A \left(C_L(\alpha^{A,b}) \frac{\partial U_n^{A,b}}{\partial x_i^a} - C_D(\alpha^{A,b}) \frac{\partial U_l^{A,b}}{\partial x_i^a} \right) \sqrt{\left(U_n^{A,b}\right)^2 + \left(U_l^{A,b}\right)^2} \\
&+ \frac{1}{2}\rho c_A \left(C_L(\alpha^{A,b}) U_n^{A,b} - C_D(\alpha^{A,b}) U_l^{A,b} \right) \frac{U_n^{A,b} \frac{\partial U_n^{A,b}}{\partial x_i^a} + U_l^{A,b} \frac{\partial U_l^{A,b}}{\partial x_i^a}}{\sqrt{\left(U_n^{A,b}\right)^2 + \left(U_l^{A,b}\right)^2}}.
\end{aligned} \tag{16}$$

$$\begin{aligned}
\frac{\partial \delta q_t^{A,b}}{\partial x_i^a} &= -\frac{1}{2}\rho c_A^2 \frac{\partial C_M(\alpha^{A,b})}{\partial \alpha^{A,b}} \frac{\partial \alpha^{A,b}}{\partial x_i^a} \left(\left(U_n^{A,b}\right)^2 + \left(U_l^{A,b}\right)^2 \right) \\
&- \rho c_A^2 C_M(\alpha^{A,b}) \left(U_n^{A,b} \frac{\partial U_n^{A,b}}{\partial x_i^a} + U_l^{A,b} \frac{\partial U_l^{A,b}}{\partial x_i^a} \right).
\end{aligned} \tag{17}$$

The partial derivatives

$$\frac{\partial C_L(\alpha^{A,b})}{\partial \alpha^{A,b}}, \; \frac{\partial C_D(\alpha^{A,b})}{\partial \alpha^{A,b}}, \; \frac{\partial C_M(\alpha^{A,b})}{\partial \alpha^{A,b}},$$

of the lift, drag and torque coefficients (for which there are no analytical expressions) are computed numerically from their curves. To evaluate (15)-(17) it remains to obtain expressions for

$$\frac{\partial \alpha^{A,b}}{\partial x_i}, \; \frac{\partial \alpha^{A,b}}{\partial u_i}, \; \frac{\partial \alpha^{A,b}}{\partial \beta}, \; \frac{\partial U_n^{A,b}}{\partial x_i}, \; \frac{\partial U_n^{A,b}}{\partial u_i}, \; \frac{\partial U_n^{A,b}}{\partial \beta}, \; \frac{\partial U_l^{A,b}}{\partial x_i}, \; \frac{\partial U_l^{A,b}}{\partial u_i}, \; \frac{\partial U_l^{A,b}}{\partial \beta}.$$

Since $(d\arctan(x)/dx) = 1/(1+x^2)$, from (5), (4) and (10) it follows

$$\begin{aligned}
\frac{\partial \alpha^{A,b}}{\partial x} &= \frac{U_l^{A,b}\frac{\partial U_n^{A,b}}{\partial x} - U_n^{A,b}\frac{\partial U_l^{A,b}}{\partial x}}{\left(U_n^{A,b}\right)^2 + \left(U_l^{A,b}\right)^2} - e_b^T T_M(\psi_k) C_\phi, && (18) \\
\frac{\partial \alpha^{A,b}}{\partial w_i} &= \frac{U_l^{A,b}\frac{\partial U_n^{A,b}}{\partial w_i} - U_n^{A,b}\frac{\partial U_l^{A,b}}{\partial w_i}}{\left(U_n^{A,b}\right)^2 + \left(U_l^{A,b}\right)^2}, \text{ with } w \doteq \begin{bmatrix} u \\ \beta \end{bmatrix} && (19)
\end{aligned}$$

where e_b denotes the b-th column of the 3×3 identity matrix. From (5), (2) and (1), and under the assumptions on the ADM model, given in Section 4.1, the partial derivatives of $U_n^{A,b}$ and $U_l^{A,b}$ with respect to the elements of the vector x^a are given by

$$\begin{aligned}
\frac{\partial U_n^{A,b}}{\partial x} &= 0, && (20) \\
\frac{\partial U_n^{A,b}}{\partial u_i} &= T_1(\beta)\delta_{b,i} + \frac{\partial \delta U_{i,corr}^{A,b}}{\partial u_i}, && (21) \\
\frac{\partial U_n^{A,b}}{\partial \beta} &= -T_2(\beta)(\bar{U} + u_b) + \frac{\partial \delta U_{i,corr}^{A,b}}{\partial \beta}, && (22) \\
\frac{\partial U_l^{A,b}}{\partial x} &= \frac{\bar{V}_l^{A,b}}{\bar{\Omega}} C_\Omega, && (23) \\
\frac{\partial U_l^{A,b}}{\partial u_i} &= \left(\sin(\psi^b)T_2(\beta) - \cos(\psi^b)T_3\right)\delta_{b,i}, && (24) \\
\frac{\partial U_l^{A,b}}{\partial \beta} &= T_1(\beta)\sin(\psi^b)(\bar{U} + u_b). && (25)
\end{aligned}$$

Finally, we need to derive expressions for $(\partial \delta U_{i,corr}^{A,b}/\partial w_i)$ to be used in (21) and (22). To this end, denote

$$\begin{aligned}
f_1 &= \tan\left(\tfrac{1}{2}\arctan\left(\tfrac{g_1 g_3}{g_2 g_3 - \bar{U}_i^{A_{2/3}}}\right)\right), \\
g_1 &= \sqrt{\cos^2(\bar{\phi}_{tlt})\sin^2(\bar{\phi}_{yw} + \beta) + \sin^2(\bar{\phi}_{tlt})}, \\
g_2 &= \cos(\bar{\phi}_{tlt})\cos(\bar{\phi}_{yw} + \beta), \\
g_3 &= \bar{U} + u^R.
\end{aligned}$$

Using (3) and under the assumption $\delta U_i = 0$ (see Section 4.1) it can easily be verified that

$$\delta U_{i,corr}^{A,b} = \frac{15\pi}{64R} r_A f_1 \cos(\psi^b - \delta\psi)\bar{U}_i^{A_{2/3}}. \qquad (26)$$

Hence,

$$\begin{aligned}
\frac{\partial g_1}{\partial \beta} &= \frac{1}{g_1}\cos^2(\bar{\phi}_{tlt})\sin(\bar{\phi}_{yw} + \beta)\cos(\bar{\phi}_{yw} + \beta), && (27) \\
\frac{\partial g_2}{\partial \beta} &= -\cos(\bar{\phi}_{tlt})\sin(\bar{\phi}_{yw} + \beta), && (28) \\
\frac{\partial g_3}{\partial u_i} &= \frac{1}{3}, && (29)
\end{aligned}$$

and using the fact that $d\tan(x)/dx = \sec^2(x)$ we have

$$
\begin{aligned}
\frac{\partial f_1}{\partial \beta} &= \sec^2\left(\frac{1}{2}\arctan\left(\frac{g_1 g_3}{g_2 g_3 - \bar{U}_i^{A_{2/3}}}\right)\right) \frac{\frac{1}{2}}{1+\left(\frac{g_1 g_3}{g_2 g_3 - \bar{U}_i^{A_{2/3}}}\right)^2} \frac{\frac{\partial g_1}{\partial \beta} g_3 (g_2 g_3 - \bar{U}_i^{A_{2/3}}) - g_1 g_3^2 \frac{\partial g_2}{\partial \beta}}{\left(g_2 g_3 - \bar{U}_i^{A_{2/3}}\right)^2} \\
\frac{\partial f_1}{\partial u_i} &= \sec^2\left(\frac{1}{2}\arctan\left(\frac{g_1 g_3}{g_2 g_3 - \bar{U}_i^{A_{2/3}}}\right)\right) \frac{\frac{1}{2}}{1+\left(\frac{g_1 g_3}{g_2 g_3 - \bar{U}_i^{A_{2/3}}}\right)^2} \frac{\frac{\partial g_3}{\partial u_i} g_1 (g_2 g_3 - \bar{U}_i^{A_{2/3}}) - g_1 g_2 g_3 \frac{\partial g_3}{\partial u_i}}{\left(g_2 g_3 - \bar{U}_i^{A_{2/3}}\right)^2}
\end{aligned}
\tag{30}
$$

Next, notice that the expression for $\delta\psi$ in equation (3) can more compactly be written as

$$
\delta\psi = \begin{cases} \frac{\pi}{2}\,\mathrm{sgn}\left(U_{tlt}^{\beta,gust}\right), & \text{if } U_{yw}^{\beta,gust} = 0 \\ \arctan\left(\frac{U_{tlt}^{\beta,gust}}{U_{yw}^{\beta,gust}}\right), & \text{if } U_{yw}^{\beta,gust} > 0 \\ \pi + \arctan\left(\frac{U_{tlt}^{\beta,gust}}{U_{yw}^{\beta,gust}}\right), & \text{if } U_{yw}^{\beta,gust} < 0 \end{cases}
$$

and since it follows from (2) that

$$
\frac{U_{tlt}^{\beta,gust}}{U_{yw}^{\beta,gust}} = \frac{T_3(\beta)}{T_2(\beta)},
$$

we have

$$
\frac{\partial \delta\psi}{\partial \beta} = \begin{cases} 0, & \text{if } T_2(\beta) = 0. \\ \frac{\partial}{\partial \beta}\arctan\left(\frac{\sin(\bar{\phi}_{tlt})}{\cos(\bar{\phi}_{tlt})\sin(\bar{\phi}_{yw}+\beta)}\right), & \text{otherwise.} \end{cases}
$$

Then, denoting

$$
f_2 \doteq \cos(\psi^b - \delta\psi)
$$

we can write (with a slight abuse of notation)

$$
\frac{\partial f_2}{\partial u_i} = 0. \tag{31}
$$

$$
\frac{\partial f_2(T_2(\beta) = 0)}{\partial \beta} = 0, \tag{32}
$$

$$
\frac{\partial f_2(T_2(\beta) \neq 0)}{\partial \beta} = -\sin(\psi^b - \delta\psi) \frac{-1}{1 + \frac{\sin^2(\bar{\phi}_{tlt})}{\cos^2(\bar{\phi}_{tlt})\sin^2(\bar{\phi}_{yw}+\beta)}} \frac{-\sin(\bar{\phi}_{tlt})\cos(\bar{\phi}_{yw}+\beta)}{\cos(\bar{\phi}_{tlt})\sin^2(\bar{\phi}_{yw}+\beta)} \tag{33}
$$

Hence, from (26) we finally have

$$
\frac{\partial \delta U_{i,corr}^{A,b}}{\partial u_i} = \frac{15\pi}{64R} r_A \frac{\partial f_1}{\partial u_i} \cos(\psi^b - \delta\psi) U_i^{A_{2/3}}, \tag{34}
$$

$$
\frac{\partial \delta U_{i,corr}^{A,b}}{\partial \beta} = \frac{15\pi}{64R} r_A \left(\frac{\partial f_1}{\partial \beta}\cos(\psi^b - \delta\psi) + f_1 \frac{\partial f_2}{\partial \beta}\right) U_i^{A_{2/3}}. \tag{35}
$$

which is the last expression needed for the computation of the derivatives in the EKF.

B Bumpless Controller Switching

Generally, when switching between two controllers large transients can occur due to the state of the controller, that is being switched on, not having a “proper” initial value. To prevent this, different methods for bumpless transfer have been proposed in the literature, such as using gradual switching based on weighted combinations between the controller being switched on and the one being switched off (Rodriguez et al., 2003), or using an inner feedback loop around the inactive controller so as to push its output to follow the one produced by the active controller (Goodwin et al., 2001; Turner and Walker, 2000). In this report, instead, an algorithm is proposed that aims directly at proper state reinitialization of the controller that is being switched on.

Although we are dealing in this report with a SISO controller for rotor speed regulation, the switching strategy is developed for the general MIMO case making it applicable to a wider class of problems. Consider a bank of controllers $\mathcal{K} = \{\mathcal{K}^{(i)}, : \; i = 1, 2, \dots, N\}$, with the j-th controller given in state-space form as follows

$$\mathcal{K}^{(j)} : \left\{ \begin{array}{rcl} x^{(j)}_{k+1} & = & A^{(j)} x^{(j)}_k + B^{(j)} u^{(ctr)}_k, \\ y^{(j)}_k & = & C^{(j)} x^{(j)}_k + D^{(j)} u^{(ctr)}_k, \end{array} \right.$$

with $x^{(j)}_k \in \mathbb{R}^{n_j}$. Notice that the controllers can have different state dimension, while they have the same input $u^{(ctr)}_k$ (filtered rotor speed error in the application of this report). No assumption is imposed on the number of inputs and outputs of the controllers.

Suppose that controller $\mathcal{K}^{(n)}$ (“n” for “new”) is being switched on at the current time instant k. The idea, used here, is to select an initial value for the controller state $x^{(n)}_k$ in such a way that a bumpless response is achieved. To this end, a past time interval $[k-N, k-1]$ of a given length N considered, for which the initial state $x^{(n)}_{k-N}$ is chosen such that using the past inputs of the previously active controller $\mathcal{K}^{(o)}$ (“o” for “old”) up to time instant $(k-1)$, $\{u^{(ctr)}_{k-N}, \dots, u^{(ctr)}_{k-1}\}$, with $\mathcal{K}^{(n)}$ an output signal is produced $\{y^{(n)}_{k-N}, \dots, y^{(n)}_{k-1}\}$ that best matches the output of $\mathcal{K}^{(o)}$, i.e. $\{y^{(o)}_{k-N}, \dots, y^{(o)}_{k-1}\}$. This hypothetical output of $\mathcal{K}^{(n)}$ is given by

$$\overbrace{\begin{bmatrix} y^{(n)}_{k-N} \\ y^{(n)}_{k-N+1} \\ \vdots \\ y^{(n)}_{k-1} \end{bmatrix}}^{Y^{(n)}_{k-N|k-1}} = \overbrace{\begin{bmatrix} C^{(n)} \\ C^{(n)} A^{(n)} \\ \vdots \\ C^{(n)} (A^{(n)})^{N-1} \end{bmatrix}}^{\mathcal{O}^{(n)}} x^{(n)}_{k-N} + \underbrace{\begin{bmatrix} D^{(n)} & & & \\ C^{(n)} B^{(n)} & D^{(n)} & & \\ \vdots & \vdots & \ddots & \\ C^{(n)} (A^{(n)})^{N-2} B^{(n)} & C^{(n)} (A^{(n)})^{N-3} B^{(n)} & \dots & D^{(n)} \end{bmatrix}}_{\Gamma^{(n)}} \underbrace{\begin{bmatrix} u^{(ctr)}_{k-N} \\ u^{(ctr)}_{k-N+1} \\ \vdots \\ u^{(ctr)}_{k-1} \end{bmatrix}}_{U^{(ctr)}_{k-N|k-1}}$$

The initial state $x^{(n)}_{k-N}$ will be computed so as to minimize the weighted discrepancy between $Y^{(n)}_{k-N|k-1}$ and the corresponding vector of outputs $Y^{(o)}_{k-N|k-1}$ that have actually been output to the system in the considered time interval, i.e.

$$x^{(n)}_{k-N} = \arg\min \left\| W \left(Y^{(n)}_{k-N|k-1} - Y^{(o)}_{k-N|k-1} \right) \right\|_2 . \tag{36}$$

The weighting matrix W can be used to put more weight on the more recent outputs. One possible selection would be to use exponential forgetting of the old data, such as

$$W = \begin{bmatrix} w_1 I & & \\ & \ddots & \\ & & w_N I \end{bmatrix}, \text{ with } w_l = e^{5\frac{l-N}{N}},\ l = 1, 2, \ldots, N,$$

where the identity matrices have all the dimension of the output signal $y_k^{(n)}$.

Clearly, the optimal solution to this least squares problem is given by

$$x_{k-N}^{(n)} = \left(W\mathcal{O}^{(n)}\right)^{\dagger} W \left(\Gamma^{(n)} U_{k-N|k-1}^{(ctr)} - Y_{k-N|k-1}^{(o)}\right),$$

where the symbol † denotes the pseudo inverse operation. However, when the extended observability matrix $\mathcal{O}^{(n)}$ is numerically rank deficient, this solution leads to numerical problems. To avoid that, a numerically robust solution is described below.

Define

$$\varepsilon \doteq W\left(Y_{k-N|k-1}^{(n)} - Y_{k-N|k-1}^{(o)}\right) = W\begin{bmatrix}\mathcal{O}^{(n)} & \Gamma^{(n)} & -I\end{bmatrix}\begin{bmatrix} x_{k-N}^{(n)} \\ U_{k-N|k-1}^{(ctr)} \\ Y_{k-N|k-1}^{(o)} \end{bmatrix},$$

and compute the QR decomposition

$$W\begin{bmatrix}\mathcal{O}^{(n)} & \Gamma^{(n)} & -I\end{bmatrix} = Q\begin{bmatrix} R_1 & R_2 & R_3 \end{bmatrix}$$

with Q being an orthogonal matrix, R_1 an upper triangular matrix, and R_1, R_2 and R_3 having the same dimension as the matrices $W\mathcal{O}^{(n)}$, $W\Gamma^{(n)}$ and W, respectively. Then obtain the numerical rank of the matrix R_1 by inspecting its main diagonal elements, and denote it as $r_o^{(n)}$. Define

$$\begin{bmatrix}\tilde{R}_1 & \tilde{R}_2 & \tilde{R}_3\end{bmatrix} \doteq \begin{bmatrix} I_{r_o^{(n)}} & 0 \end{bmatrix}\begin{bmatrix} R_1 & R_2 & R_3 \end{bmatrix}.$$

Then the optimization problem in (36) is equivalent to the least squares problem

$$\begin{aligned} \min \|\varepsilon\|_2 &= \min \|Q^T \varepsilon\|_2 \\ &= \min \left\| \begin{bmatrix} R_1 & R_2 & R_3 \end{bmatrix} \begin{bmatrix} x_{k-N}^{(n)} \\ U_{k-N|k-1}^{(ctr)} \\ Y_{k-N|k-1}^{(o)} \end{bmatrix} \right\|_2 \\ &\approx \min \left\| \begin{bmatrix} \tilde{R}_1 & \tilde{R}_2 & \tilde{R}_3 \end{bmatrix} \begin{bmatrix} x_{k-N}^{(n)} \\ U_{k-N|k-1}^{(ctr)} \\ Y_{k-N|k-1}^{(o)} \end{bmatrix} \right\|_2, \end{aligned}$$

so that

$$x_{k-N}^{(n)} = -(\tilde{R}_1)^{\dagger} \begin{bmatrix} \tilde{R}_2 & \tilde{R}_3 \end{bmatrix} \begin{bmatrix} U_{k-N|k-1}^{(ctr)} \\ Y_{k-N|k-1}^{(o)} \end{bmatrix}.$$

Finally, given the so-computed state of $\mathcal{K}^{(o)}$ at the beginning of the considered past interval of time, the state at the current time instant k is computed as

$$\begin{aligned} x_k^{(n)} &= (A^{(n)})^N x_{k-N}^{(n)} + \overbrace{\left[(A^{(n)})^{N-1}B^{(n)} \quad (A^{(n)})^{N-2}B^{(n)} \quad \dots \quad B^{(n)}\right]}^{\Gamma_N^{(n)}} U_{k-N|k-1}^{(ctr)} \\ &= -\left[(A^{(n)})^N(\tilde{R}_1)^\dagger \tilde{R}_2 + \Gamma_N^{(n)}, \quad (A^{(n)})^N(\tilde{R}_1)^\dagger \tilde{R}_3\right] \begin{bmatrix} U_{k-N|k-1}^{(ctr)} \\ Y_{k-N|k-1}^{(o)} \end{bmatrix}. \end{aligned}$$

C Blade effective wind speed approximation in case of oblique wind inflow

In van Engelen and Schaak (2007) an efficient algorithm is proposed for computing realizations of blade effective wind speeds based on interpolation between six helices (three blade-related, and three intermediate). The blade-related helices contain the blade effective wind speeds in case of constant rotor speed, constant wind speed and non-oblique stream. In case of oblique inflow, each blade effective wind speed is formed by interpolating between the corresponding blade-related helix and its nearest neighboring (intermediate) helix. This method assumes constant wind speed, constant rotor speed, not tilt misalignment and is applicable to only relatively small yaw angles (up to approx. 15 deg), as otherwise the blade gets beyond the neighboring helix, as depicted on Figure 11 where the rotor is yawed at 40 degrees, so that the first blade at azimuth 0 deg does not lie between its helix 1 and the neighboring helix 2.

Here this method is generalized to the case when the rotor has arbitrary (still in between $[-\frac{\pi}{2}, \frac{\pi}{2}]$) yaw and tilt angles, as well as varying mean wind speed (as caused by wind gusts) and varying rotor speed.

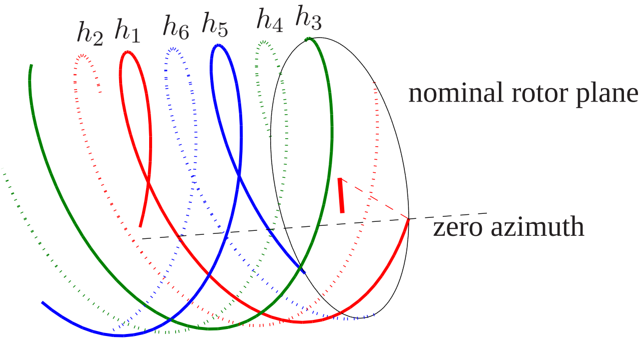


Figure 11: Interpolation based on six helices is used to approximate the blade effective wind speed in case of oblique wind flow

Definitions and assumptions

The orientations of the rotor fixed frame axes is conform the notation used in the software TURBU (van Engelen, 2007), i.e. in the case of non-oblique inflow the x axis is perpendicular to the rotor plane and is positive downwind, the z axis points downwards and the y-axis points to the right as seen from a point on the negative x-axis (upwind). In case of oblique inflow (or, equivalently, tilted and yawed rotor), the rotor fixed reference frame $(0, x_r, y_r, z_r)$ is rotated with respect to the nominal (non-oblique) rotor fixed reference frame $(0, x, y, z)$ as visualized on Figure 13. The rotor yaw angle ϕ_{yw} is defined as the angle between the y-axis and y_r-axis, measured from y to y_r in anti-clockwise direction as seen from a point on the positive z-axis. The rotor tilt angle, ϕ_{tlt}, on the other hand, is the angle between the z-axis and the z_r-axis in *clockwise* direction as seen from the positive y_r-axis.

The turbulent wind flow is assumed to have only a longitudinal component. It is further assumed that a turbulence realization on six helixes is given, denoted as $h_i(\psi_i^{hx})$ for helix $i = 1, 2, \ldots, 6$, where ψ_i^{hx} is the helix azimuth angle. The helixes are computed under the assumption of constant rotor speed $\bar{\Omega}$ and wind speed $\bar{U}$, so that under the Taylor's frozen wave hypothesis the helix azimuth angle explicitly defines a fixed point in time and space. The helixes are numbered anti-

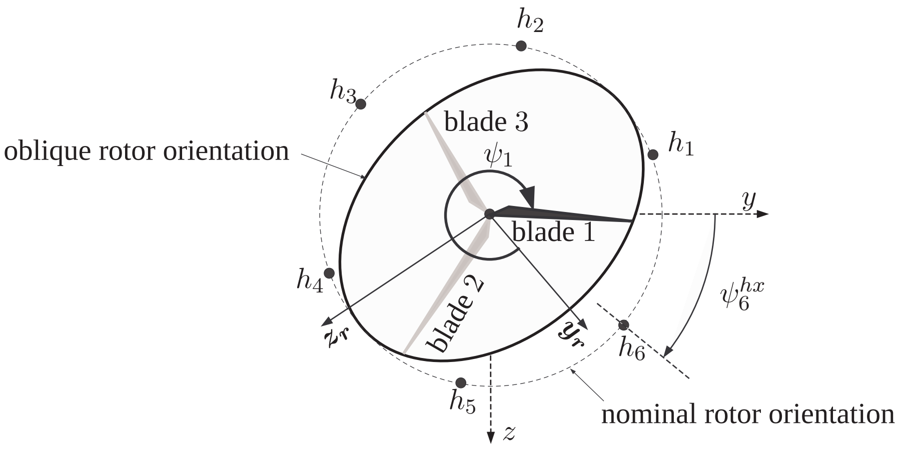


Figure 12: Visualization of tilted and yawed rotor, blade numbering, and helix numbering.

clockwise, while the rotor blades - clockwise (see Figure 12). The azimuthal angles are measured clockwise starting from the positive y_r-axis. The helixes are azimuthally equally spaced over the rotor plane. Table 4 summarizes some of the notation used. The helix azimuth angle $\psi_i^{hx}(t)$ at

notation	description
$\psi_i(t)$	azimuth of blade i
$\psi(t) \equiv \psi_1(t)$	rotor azimuth
$\psi_i^{hx}(t)$	azimuth of helix i
$U(t)$	wind speed
$\bar{U}$	wind speed used in helix realization
$\Omega(t)$	rotor speed
$\bar{\Omega}$	rotor speed used in helix realization
$\phi_{tlt}(t)$	rotor tilt angle
$\phi_{yw}(t)$	rotor yaw angle

Table 4: Definitions of symbols.

a given time instant t is defined as the azimuthal position of the intersection point of helix h_i with the *nominal* (non-oblique) rotor plane, and is hence independent on the rotor orientation (see angle ψ_6^{hx} on Figure 12). For an algorithm for computing realizations of such turbulent wind helixes, see van Engelen and Schaak (2007).

Finally, for a vector $v \in \mathbb{R}^n$, the notation v_i is used to denote the i-th element of v.

The interpolation algorithm

Given the rotor speed $\Omega(t)$ and the initial rotor position $\psi(0) = \psi_1(0)$, the azimuth angles of the rotor and the blades at time instant t are given by

$$\begin{aligned} \psi(t) &= \psi_1(0) + \int_0^\infty \Omega(t)dt, \\ \psi_b(t) &\doteq \psi(t) + \frac{2\pi(b-1)}{3},\ b = 1, 2, 3. \end{aligned} \tag{37}$$

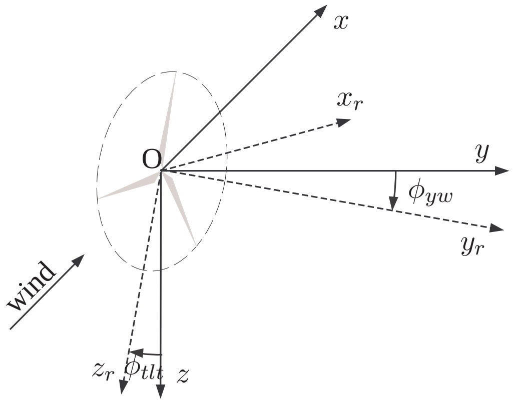


Figure 13: Nominal $(0, x, y, z)$ and oblique $(0, x_r, y_r, z_r)$ reference frames, tilt and yaw angle definition and orientation.

The helix azimuth positions at time t, on the other hand, depend on the wind speed $U(t)$ and the initial azimuth angle of the first helix $\psi_1^{hx}(0)$:

$$\begin{aligned} \psi_1^{hx}(t) &= \psi_1^{hx}(0) + \frac{\bar{\Omega}}{\bar{U}} \int_0^{\infty} U(t) dt, \\ \psi_i^{hx}(t) &\doteq \psi_1^{hx}(t) - \frac{\pi(i-1)}{3}, \; i = 1, 2 \ldots, 6. \end{aligned} \tag{38}$$

Assuming rigid rotor for simplicity of the presentation, for a given blade, say b, a point lying at distance $\frac{2R}{3}$ from the blade root[2] has the following coordinates in $(0, x_r, y_r, z_r)$ at time t

$$p_{r,b}(t) = \begin{bmatrix} 0 \\ \cos(\psi_b(t)) \\ \sin(\psi_b(t)) \end{bmatrix} \frac{2R}{3}, \; b = 1, 2, 3. \tag{39}$$

where R is the rotor radius. The coordinates of the same point in the non-oblique coordinate system $(0, x, y, z)$ can be computed using the following transformation matrices

$$\begin{aligned} P_{tlt}(\phi) &\doteq \begin{bmatrix} \cos(\phi) & 0 & \sin(\phi) \\ 0 & 1 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) \end{bmatrix}, \\ P_{yw}(\phi) &\doteq \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

that represent rotations around the z-axis (yaw) and around the y-axis (tilt) in the defined directions. Therefore, the coordinates of the $\frac{2R}{3}$ point on blade b in reference frame $(0, x, y, z)$ are given by

$$p_b(t) = \begin{bmatrix} x_b(t) \\ y_b(t) \\ z_b(t) \end{bmatrix} = P_{tlt}\big(-\phi_{tlt}(t)\big) P_{yw}\big(-\phi_{yw}(t)\big) p_{r,b}(t), \; b = 1, 2, 3. \tag{40}$$

[2] This point is assumed to be the effective location for taking into account the blade position relative to the helixes.

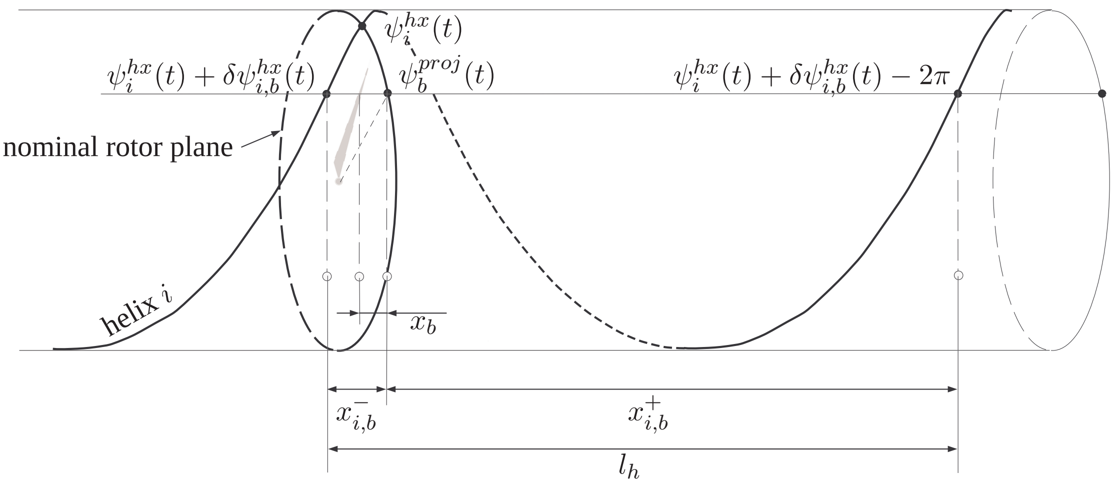


Figure 14: Visualization of the defined azimuth angles and lengths.

Hence, the *projected* blade onto the nominal rotor plane has azimuth (modulus 2π)

$$\psi_b^{proj}(t) = \begin{cases} \pi - \arcsin\left(\frac{|z_b(t)|}{\sqrt{y_b^2(t)+z_b^2(t)}}\right), & \text{if } y_b(t), < 0 \text{ and } z_b(t) > 0 \\ \pi + \arcsin\left(\frac{|z_b(t)|}{\sqrt{y_b^2(t)+z_b^2(t)}}\right), & \text{if } y_b(t) < 0 \text{ and } z_b(t) < 0 \\ 2\pi - \arcsin\left(\frac{|z_b(t)|}{\sqrt{y_b^2(t)+z_b^2(t)}}\right), & \text{if } y_b(t) > 0 \text{ and } z_b(t) < 0 \\ \arcsin\left(\frac{|z_b(t)|}{\sqrt{y_b^2(t)+z_b^2(t)}}\right), & \text{if } y_b(t) > 0 \text{ and } z_b(t) > 0 \end{cases} \tag{41}$$

For helix i, the difference between the helix azimuth $\psi_i^{hx}(t)$ and the projected azimuth of blade b is then

$$\delta\psi_{i,b}^{hx}(t) = (\psi_b^{proj}(t) - \psi_i^{hx}(t)) \mod (2\pi). \tag{42}$$

Figure 14 depicts the helix tube on which the six helixes lie, as well as the nominal rotor plane, and some azimuth angles and lengths, needed in the sequel. At time instant t, the $\frac{2R}{3}$ point on blade b lines on the azimuth line through $\psi_b^{proj}(t)$, which line intersects with helix i at infinitely many points, but the closest two to the nominal rotor plane correspond to helix i azimuth angles $(\psi_i^{hx}(t) + \delta\psi_{i,b}^{hx}(t))$ and $(\psi_i^{hx}(t) + \delta\psi_{i,b}^{hx}(t) - 2\pi)$. In reference frame $(0, x, y, z)$, these two points have cetrain x-coordinates $x_{i,b}^-(t)$ and $x_{i,b}^+(t)$. Given that the helix is generated under the assumption of constant wind speed and rotor speed, the helix length is given by

$$l_h = \bar{U}\frac{2\pi}{\bar{\Omega}}, \tag{43}$$

so that

$$\begin{aligned} x_{i,b}^-(t) &= \frac{-\delta\psi_{i,b}^{hx}(t)}{2\pi} l_h, \\ x_{i,b}^+(t) &= l_h + x_{i,b}^-(t). \end{aligned} \tag{44}$$

Given the current x-position of the $\frac{2R}{3}$ point of blade b at time instant t, $x_b(t)$, the next thing to do is to determine the closest two helixes, so as to subsequently interpolate between them. To this end, define the matrices

$$X^{(b)}(t) \doteq \begin{bmatrix} x_{1,b}^{-}(t) \\ \vdots \\ x_{6,b}^{-}(t) \\ x_{1,b}^{+}(t) \\ \vdots \\ x_{6,b}^{+}(t) \end{bmatrix}, H^{(b)}(t) \doteq \begin{bmatrix} h_1(\psi_1^{hx}(t) + \delta\psi_{1,b}^{hx}(t)) \\ \vdots \\ h_6(\psi_6^{hx}(t) + \delta\psi_{6,b}^{hx}(t)) \\ h_1(\psi_1^{hx}(t) + \delta\psi_{1,b}^{hx}(t) - 2\pi) \\ \vdots \\ h_6(\psi_6^{hx}(t) + \delta\psi_{6,b}^{hx}(t) - 2\pi) \end{bmatrix}. \tag{45}$$

Then the closest helix in downwind direction is $H_{i_b^{dn}}^{(b)}(t)$ with

$$i_b^{dn} = \arg\min_i \left\{ X_i^{(b)}(t) - x_b(t) : \ X_i^{(b)}(t) \geq x_b(t) \right\}, \ b = 1,2,3. \tag{46}$$

Similarly, upwind the closest helix is $H_{i_b^{up}}^{(b)}(t)$ with

$$i_b^{up} = \arg\min_i \left\{ x_b(t) - X_i^{(b)}(t) : \ X_i^{(b)}(t) \leq x_b(t) \right\}, \ b = 1,2,3. \tag{47}$$

Notice that the indexes i_b^{dn} and i_b^{up} are also time depended, although not explicitly denoted.

Then a linear interpolation is performed based on the distances between the blade point and the closest helixes. This is done by defining the interpolation weighting factor

$$\alpha_b(t) = \frac{6\left(X_{i_b^{dn}}^{(b)}(t) - x_b(t) \right)}{l_h} \in [0,1], \ b = 1,2,3, \tag{48}$$

so that the following convex combination between the two selected helixes can be used

$$\tilde{u}_b(t) = (1 - \alpha_b(t)) H_{i_b^{dn}}^{(b)}(t) + \alpha_b(t) H_{i_b^{up}}^{(b)}(t), \ b = 1,2,3.$$

Compensation for the covariance of $\tilde{u}_b(t)$

In the above expression for $\tilde{u}_b(t)$, a convex combination is taken between two stochastic signals, namely $H_{i_b^{dn}}^{(b)}(t)$ and $H_{i_b^{up}}^{(b)}(t)$. Assuming stationary homogeneous turbulence field with spectrum at (any) fixed point in space $S_u(w)$, and denoting $\tilde{u}(t)$ as the turbulence at (any) fixed point in space at time t, the following two expressions hold for the first two moments of $u_b(t)$

$$\begin{aligned} &E\{\tilde{u}_b(t)\} = E\{\tilde{u}(t)\} = 0, \\ &E\{\tilde{u}_b^2(t)\} = ((1-\alpha_b(t))^2 + \alpha_b^2(t)) \underbrace{E\{\tilde{u}^2(t)\}}_{\sigma} + 2\alpha_b(t)(1-\alpha_b(t)) \underbrace{E\{H_{i_b^{dn}}^{(b)}(t) H_{i_b^{up}}^{(b)}(t)\}}_{c(d,0)}, \end{aligned}$$

where σ denotes the variance of a fixed point in space, while $c(d,\tau)$ is the covariance function between two fixes points in space at a distance $d = \left(X_{i_b^{dn}}^{(b)}(t) - X_{i_b^{up}}^{(b)}(t) \right)$. Hence, the variance of $\tilde{u}_b(t)$ is not equal to the turbulence variance σ. In order to make the two variances the same, an additional covariance correction factor, $\rho_b(t)$, will be used, so that $E\{(\rho_b(t)\tilde{u}_b(t))^2\} = \sigma$.

Then denoting $\gamma(d,\omega)$ as the coherence function between any two points in space at a distance d

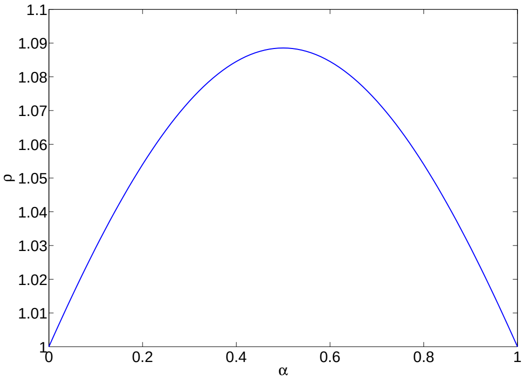


Figure 15: Plot of the covariance correction factor ρ as a function of the interpolation weighting factor α.

one has

$$\begin{aligned} \sigma &= \int_{-\infty}^{\infty} S_u(\omega)d\omega, \\ c\,(d,0) &= \int_{-\infty}^{\infty} \gamma\,(d,\omega)\,S_u(\omega)d\omega. \end{aligned} \tag{49}$$

Then, since the distance, $\left(X_{i_b^{dn}}^{(b)}(t) - X_{i_b^{up}}^{(b)}(t)\right)$, between the (neighboring) helixes $H_{i_b^{dn}}^{(b)}(t)$ and $H_{i_b^{up}}^{(b)}(t)$ is exactly $\frac{1}{6}l_h$, it can easily be verified that

$$\rho_b(t) = \frac{1}{\sqrt{(1-\alpha_b(t))^2 + \alpha_b^2(t) + 2\alpha_b(t)(1-\alpha_b(t))\frac{c\left(\frac{l_h}{6},0\right)}{\sigma}}},\ b = 1,2,3. \tag{50}$$

achieves $E\{(\rho_b(t)\tilde{u}_b(t))^2\} = \sigma$. The parameter ρ as a function of the interpolation factor α is depicted on Figure 15 for the following specific choices for the spectrum $S_u(\omega)$ and coherence $\gamma(d,\omega)$

$$\begin{aligned} S_u(\omega) &= \frac{\sigma_w^2 2L_1/\bar{U}}{\left(1+6L_1\omega/(2\pi\bar{U})\right)^{5/3}} \text{ (Kaimal spectrum)}, \\ \gamma(d,\omega) &= e^{-8.8d\sqrt{(\omega/(2\pi\bar{U}))^2+(0.12\Lambda/3.5)^2}}, \end{aligned}$$

with $\bar{U} = 15$ m/s, $\sigma_w = \frac{I_{15}(15+a\bar{U})}{a+1}$, $I_{15} = 0.17$, $a = 3$, $L_1 = 170.1$ m, and $\Lambda = 21$ m. Hence, adding the rotor-wide wind speed $U(t)$ to the *corrected* expression for the turbulence $(\rho_b(t)\tilde{u}_b(t))$, the final expression for the blade-effective wind speed takes the form

$$u_b(t) = U(t) + \rho_b(t)\left((1-\alpha_b(t))H_{i_b^{dn}}^{(b)}(t) + \alpha_b(t)H_{i_b^{up}}^{(b)}(t)\right),\ b = 1,2,3. \tag{51}$$

Numerical implementation

The complete algorithm for approximation of blade effective wind speeds under oblique wind inflow conditions consists of evaluation of the expressions in equations (37)-(51) at each time instant t and for each blade b. In practice, the rotor speed and the azimuth angle of the rotor is measured, so (37) need not be numerically evaluated.

In a numerical implementation the same steps can be followed at discrete time instants (kt_s),

$k = 0, 1, \ldots$, after making the following small modifications:

Equation (38) Assuming that the wind speed does not change between any two time instants, i.e. $U(kt_s + \tau) = U(kt_s)$ for $\tau \in [0, t_s)$, the expression for the azimuth of the first helix takes the form

$$\psi_1^{hx}(kt_s) = \psi_1^{hx}(0) + \frac{\bar{\Omega}}{\bar{U}} \sum_{l=0}^{k-1} U(lt_s)t_s.$$

A better option would to use more advanced numerical integration methods to compute $\psi_1^{hx}(kt_s)$.

Equation (45) In a numerical implementation the helixes are only given at discrete azimuth angles, so that it is in general not possible to evaluate $H^{(b)}(t)$ at the desired azimuth angles. One way to circumvent this problem is to evaluate $H^{(b)}(t)$ instead at the closest azimuth angles at which the helixes are given. Assuming that helix i is defined at azimuth angles $(\psi_i^{hx}(0) + k\delta\psi^{hx})$, $k = 0, 1, \ldots$, and define the following projection

$$\begin{aligned}
\Pi_i(x) &\doteq \psi_i^{hx}(0) + \left(\arg\min_k \left|\psi_i^{hx}(0) + k\delta\psi^{hx} - x\right|\right) \delta\psi^{hx} \\
&\overset{(38)}{=} \psi_1^{hx}(0) - \tfrac{\pi(i-1)}{3} + \left(\arg\min_k \left|\psi_1^{hx}(0) - \tfrac{\pi(i-1)}{3} + k\delta\psi^{hx} - x\right|\right) \delta\psi^{hx} \\
&= \psi_1^{hx}(0) - \frac{\pi(i-1)}{3} + \text{round}\left(\frac{x - \psi_1^{hx}(0) + \frac{\pi(i-1)}{3}}{\delta\psi^{hx}}\right) \delta\psi^{hx}
\end{aligned}$$

that maps x onto the set of azimuth angles at which helix i is defined. In this way, the expression for $H^{(b)}(t)$ in (45) should simply be replaced by

$$H^{(b)}(t) = \begin{bmatrix} h_1\left(\Pi_1(\psi_1^{hx}(t) + \delta\psi_{1,b}^{hx}(t))\right) \\ \vdots \\ h_6\left(\Pi_6(\psi_6^{hx}(t) + \delta\psi_{6,b}^{hx}(t))\right) \\ h_1\left(\Pi_1(\psi_1^{hx}(t) + \delta\psi_{1,b}^{hx}(t) - 2\pi)\right) \\ \vdots \\ h_6\left(\Pi_6(\psi_6^{hx}(t) + \delta\psi_{6,b}^{hx}(t) - 2\pi)\right) \end{bmatrix}.$$

Numerical example

The algorithm is numerically tested with the data given in Table 5. The helixes are generated based on the assumption of Kaimal fixed point turbulence spectrum, and under an extreme wind condition, occurring at $t = 5$ s, and comprising of a rising wind gust of 15 m/s in combination of a yaw angle of 30 degrees. The wind gust and the yaw angle are given on Figure 16 as functions of time. On each plot in Figure 17 there are four lines. The three dashed lines on all three plots are the same and correspond to the three blade related helixes (helixes 1,5 and 3); these coincide with the blade effective wind speeds in the case of non-oblique inflow and constant rotor and wind speeds. The other three helixes are not plotted. The solid lines on the plots represent the blade effective wind speeds as computed by the proposed algorithm, one per plot.

symbol	value	description
$\bar{U}$	15 m/s	mean wind speed
$U(t)$	see Figure 16 (left)	wind gust
$\bar{\Omega}$	1.85 rad/s	mean rotor speed
$\Omega(t)$	$\bar{\Omega}$	rotor speed
$\psi(0)$	354.7 deg	initial rotor azimuth
$\psi_1^{hx}(0)$	-5.3 deg	initial azimuth helix 1
t_s	0.02 s	sampling time
$\delta\psi^{hx}$	5.3 deg	helix azimuth sampling angle
$\phi_{yw}(t)$	see Figure 16 (right)	rotor yaw angle
$\phi_{tlt}(t)$	-5.1271 deg	rotor tilt angle
$c(l_h/6,0)/\sigma$	0.6879	parameter in equation (50)

Table 5: Data used in the numerical example

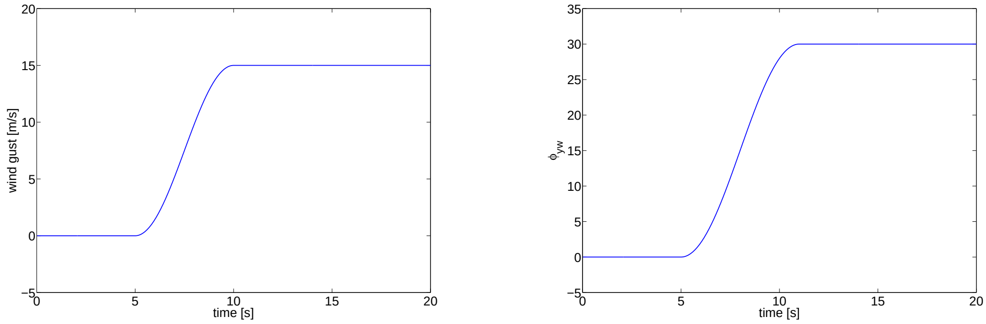


Figure 16: Wind gust $U(t)$ (left) and yaw angle $\phi_{yw}(t)$ (right).

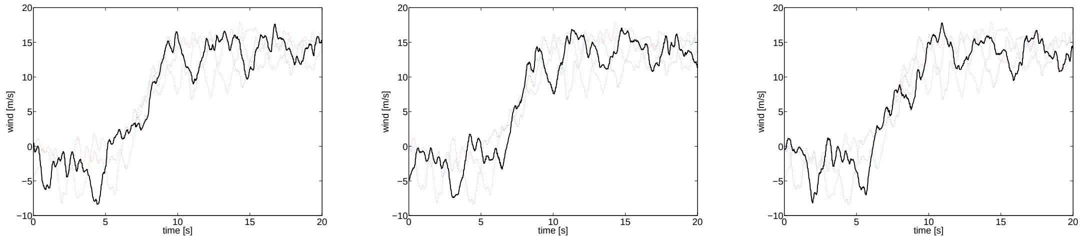


Figure 17: Helixes 1,3 and 5 (dashed lines) and the blade effective wind speeds (solid) of blades 1 (left), 2 (middle) and 3 (right).