1. INTRODUCTION

Extreme wind conditions, such as wind gusts and/or wind direction changes, can lead to very large turbine loads causing fatigue, automatic shutdowns 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 (EEC)]. In this paper, the extreme wind gust and direction change recognition (EG&DR) 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 activate an 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,\textsuperscript{1,2} or by considering a simple first-order model of the rotor dynamics (i.e. neglecting shaft torsion).\textsuperscript{3,5} Recently, somewhat more advanced models have been used, including first shaft torsion mode to the rotor dynamics.\textsuperscript{6} 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 unmodeled higher-order dynamics such as tower sideward 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 been studied, including extended Kalman filter (EKF),\textsuperscript{2} linear Kalman filter in combination with $T_a$-tracking control loop\textsuperscript{6} or augmented-state non-linear filters.\textsuperscript{5} Still, all these publications have

RESEARCH ARTICLE

Wind turbine extreme gust control
Stoyan Kanev and Tim van Engelen
Energy Research Center of the Netherlands, Unit Wind Energy, 1755ZG Petten, The Netherlands

ABSTRACT

This paper focuses on the problem of extreme wind gust and direction change recognition (EG&DR) and control (EEC). An extreme wind gust with direction change can lead to large loads on the turbine (causing fatigue) and unnecessary turbine shutdows by the supervisory system caused by rotor overspeed. The proposed EG&DR algorithm is based on a non-linear 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 (cumulative sum test) to recognize extreme events. The non-linear 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 $H\infty$ optimal control setting. The method is demonstrated on a complex non-linear test turbine model. Copyright © 2009 John Wiley & Sons, Ltd.

KEYWORDS

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

Correspondence

S. Kanev, Energy Research Center of the Netherlands, Unit Wind Energy, PO Box 1, 1755ZG Petten, The Netherlands.
E-mail: kanev@ecn.nl

Received 1 February 2008; Revised 23 January 2009; Accepted 29 March 2009

WIND ENERGY
Published online 5 May 2009. DOI: 10.1002/we.338
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 EG&DR algorithm developed in this paper. More specifically, an augmented state EKF is utilized, based on a non-linear wind turbine model. This model consists of a linear structural dynamics model (SDM) on which aerodynamic forces and torques are acting as computed by a non-linear 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 1 (in which the physical meaning of the signals is described later on). These components are:

- The 40th order linearized structural dynamics model (SDM), obtained using the software TURBU,\(^7\) with degrees of freedom in tower foundation, blade flanges and drive train, and including pitch actuator dynamics;
- Non-linear ACM based on blade element momentum (BEM) theory, including: (i) dynamic wake effects as modeled by the ECN differential equation model;\(^6\) (ii) Glauert’s azimuth-dependent correction term for the axial induction speed in case of oblique inflow;\(^5\) and (iii) correction on the angle of attack caused by rotor coning, as implemented in the non-linear aerelastic wind turbine simulation tool PHATAS;\(^10\)
- 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;
- Non-linear generator torque controller based on static optimal-\(\lambda\) 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 non-linearities 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 1; and
- Realistic blade-effective wind speed signals are generated based on the helix approximation concept, as proposed in Kanev and van Engelen,\(^11\) App. C, 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 aerodynamic module (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 switched on again when either the (filtered) rotor speed begins decreasing, or the pitch angles have reached a suitable defined reference value, which is a function of the axial component of the (estimated) wind speed. The last one is computed offline 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 load reduction control should only be 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 paper, the IPC is only active whenever the estimated oblique inflow angle is larger (in absolute value) than 10°.

The paper is organized as follows. The next section explains the notation used throughout the paper, as well as the physical meaning of the used variables. ‘Turbine Simulation Model’ describes the structure and the main components of the turbine simulation model. The algorithm for detection of extreme events is developed in ‘Extreme Event Recognition’, while EEC is the topic of ‘Extreme Event Control’. The complete EG\&DR–EEC method is tested in simulations in ‘Simulation’. The paper is concluded in ‘Conclusion’.

2. NOTATION AND SYMBOLS

For a scalar or vector variable \( v \), \( v' \) denotes its equilibrium or mean value, while \( \delta v = v - v' \) is called the (current) variation around the equilibrium value. A superscript \( cm \), as in \( v^{cm} \), means that the variable is defined in multiblade coordinates as obtained by performing a Coleman demodulation (see ‘Simplified Model’) of the signal \( v \) (\( v \) being defined in the rotating reference frame). Subscripts/subscripts \( b \) and \( A \), as in \( U^b, U^A \), denote the number of the blade \( (b = 1, 2, 3) \) and the number of the blade element \( (A = 1, 2, \ldots, N_{\text{elem}}) \) 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 vec(\( A \)) stacks the columns of the matrix \( A \) below each other into one vector. The operator \( \odot \) represents the direct sum of matrices, i.e. \( A \odot B = \text{blockdiag}(A, B) \). The \( n \)-by-\( n \) identity matrix is denoted as \( I_n \) and \( \delta_{ik} \) is the Kronecker delta function.

The following symbols (with SI dimensions) are used in the text:

- \( c_A \): cord length of blade element \( A \)
- \( C_L, C_D, C_M \): lift, drag and pitch-wise torque coefficients
- \( M_b, M_f \): lead-wise (in-plane) and flap-wise (out-of-plane) blade root bending moment
- \( M_L, M_C \): lead-wise (in-plane) and flap-wise (out-of-plane) blade root bending moment
- \( P_k \): state covariance matrix in the EKF
- \( q_i \): aerodynamic pitch-wise moment (nose-down positive) of element \( A \) of blade \( b \)
- \( q_i^{ab} \): aerodynamic forces in normal and lead-wise direction of element \( A \) of blade \( b \)
- \( R \): rotor radius
- \( r_A \): distance from hub center to center of blade element \( A \)
- \( T_g \): generator torque reference (output of controller)
- \( T_s, T^s \): sample time turbine model, sample time of controller
- \( \bar{U} \): mean undisturbed wind speed in the longitudinal wind field direction
- \( \bar{U}_{\text{act}}, \bar{U}_{\text{yr}}, \bar{U}_{\text{in}} \): axial, yaw-oriented and tilt-oriented components of \( \bar{U} \)
- \( \bar{U}_i^A \): equilibrium axial and tangential induction wind speeds
- \( \delta U_i^A \): dynamic term on the axial induction wind speed
- \( \delta U_{i,\text{cor}}^A \): Glauert’s correction term to \( U_i^A \) for oblique inflow
- \( U_{i,\text{vn}}^A \): axial induction wind speed of annulus at \( 2/3R \)
- \( u_b, \bar{V}_{\text{in}}^A, \bar{V}_{\text{vn}}^A \): blade \( b \) effective wind speed, equilibrium normal and lead-wise effective wind speed at blade element \( A \)
- \( x, x' \): normal and lead-wise effective wind speed variation at element \( A \) of blade \( b \)
- \( \alpha^{ab} \): angle of attack of element \( A \) of blade \( b \)
- \( \beta \): additional (to \( \phi \)) yaw misalignment angle for modeling wind direction change
- \( \theta_b \): pitch angle reference for blade \( b \) (output of controller)
- \( \rho \): air density
- \( \phi^{ab} \): pitch angle of element \( A \) of blade \( b \)
- \( \phi_{\text{yr}}, \phi_{\text{lt}} \): equilibrium yaw and tilt angles of the wind speed \( \bar{U} \) (see Figure 2)
- \( \psi' \): azimuth angle of blade \( b \), rotor azimuth
- \( \delta \psi \): azimuth offset angle caused by oblique inflow orientation
- \( \Omega, \Omega' \): rotor speed, filtered rotor speed

3. TURBINE SIMULATION MODEL

The turbine simulation model represents a typical three-bladed horizontal-axis wind turbine (HAWT). The model consists of an integration of several blocks, as sketched in Figure 1. 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. The model assumes rigid blades and tower, but contains degrees of freedom in the blade flanges, tower foundation and rotor shaft, and includes the pitch actuator dynamics. Although the blades are considered rigid, there are \( N_{\text{ann}} = 14 \) blade elements per blade, allowing for a better representation of the aerodynamic forces, as computed from the ADM block, described in ‘ADM’. The model (see Figure 1) has: (i) 40 states: positions and speeds in three directions for the three-blade flange elements and the tower bottom element, rotational position and speed for the two drive-train elements and four states per blade for modeling the servo-pitch actuators at the three blades (all states defined in multiblade coordinates, see ‘Simplified Model’); (ii) 130 inputs: three reference blade pitch angles \( \theta^{\text{cm}} \), one reference generator torque \( T_g \), three \( N_{\text{ann}} \) blade element torques \( q_{i,n}^{\text{cm}} \), three \( N_{\text{ann}} \) normal forces \( q_{i,n}^{\text{nm}} \) and three \( N_{\text{ann}} \) leadwise forces \( q_{i,l}^{\text{cm}} \), all in multiblade coordinates; and (iii) 133 outputs: rotor speed \( \Omega \), three blade root out-of-plane bending moments \( M_{z}^{\text{cm}} \), three blade root in-plane bending moments \( M_{x}^{\text{cm}} \), three \( N_{\text{ann}} \) blade element pitch angles \( (\delta \phi_{i}^{a,b})^{\text{cm}} \), three \( N_{\text{ann}} \) normal velocities \( (\delta V_{n}^{a,b})^{\text{cm}} \) and three \( N_{\text{ann}} \) lead-wise velocities \( (\delta V_{l}^{a,b})^{\text{cm}} \), also in multiblade coordinates.

The inputs \( \theta^{\text{cm}} \) and \( T_g \) are controlled inputs, the outputs \( \Omega, M_{z}^{\text{cm}} \) and \( M_{x}^{\text{cm}} \) 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 \( u_h \) 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 in Kanev and van Engelen (App. C). These blade-effective wind speeds are computed in such a way that the resulting flap-wise 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 \( \beta \) is generated by the wind generation module, which represents yawed flow.

3.3. ADM

Because of page limitation, only a summary of the ADM algorithm is given here. For details, see Kanev and van Engelen.

3.3.1. Algorithm 3.1 (ADM)

Equilibrium values and parameters from TURBU: \( \bar{U}_{\text{ax}}, \bar{U}_{\text{yw}}, \bar{U}_{\text{tlt}}, \bar{V}_{\text{ax}}, \bar{V}_{\text{yw}}, \bar{V}_{\text{tlt}}, \bar{\phi}_{i}^{a,b}, \bar{q}_{i,n}^{a,b}, \bar{q}_{i,l}^{a,b}, \bar{q}_{i,l}^{b}, r_{k}, c_{e}, R, \rho, C_{t}(\alpha), C_{l}(\alpha), C_{D}(\alpha) \).

From SDM and wind module: \( \psi, \delta \phi^{a,b}, \delta V^{a,b}, \delta V_{n}^{a,b}, \beta, u_h \)

From ADM at previous time instant: \( \partial U_{t}^{l} \)

Step 1 Compute the undisturbed wind speeds in axial, yaw and tilt orientation, including turbulence and wind gusts contained in the blade-effective wind speed variations \( u_{g} \):

\[
\begin{bmatrix}
U_{\text{in}}^{U}& U_{\text{in}}^{U_{g}}& U_{\text{in}}^{V_{g}}
\end{bmatrix}
= \begin{bmatrix}
\cos(\hat{\phi}_{n})\cos(\hat{\phi}_{l} + \beta) \\
\cos(\hat{\phi}_{n})\sin(\hat{\phi}_{l} + \beta) \\
\sin(\hat{\phi}_{l})
\end{bmatrix}
\begin{bmatrix}
\bar{U} + \frac{1}{2} \sum u_{h}
\end{bmatrix}
\]  \( (1) \)
**Step 2** Compute Glauc’s correction $\delta U_{i,com}^{A,b}$ to the axial induction speed

$$\delta U_{i,com}^{A,b} = \frac{15\pi}{64R} \tan \left( \arctan \left( \frac{\sqrt{(U_{e,x}^{gust} - U_{s}^{gust})^2 + (U_{l,com}^{gust} - U_{s}^{gust})^2}}{U_{e,x}^{gust} - U_{s}^{gust}} \right) \right) \cos(\psi^b - \delta \psi) U_{i,com}^{A,b}$$

**Step 3** Compute setting angles of blade elements $\phi^{i,b}$, including angle of attack correction caused by rotor coning.

\[
\begin{bmatrix}
\delta \phi_i^{A,b} \\
\delta \phi_n^{A,b} \\
\delta \phi_w^{A,b}
\end{bmatrix} = \begin{bmatrix}
\cos(\delta \phi_n) \cos(\delta \phi_w + \beta) \\
\cos(\delta \phi_n) \sin(\delta \phi_w + \beta) \\
\sin(\delta \phi_n)
\end{bmatrix} \left[ u_b - \frac{1}{3} \sum_{b=1}^{b=3} u_b \right]
\]

\[
U_{n}^{A,b} = U_{e,x}^{gust} - U_{s}^{A} - \delta U_{n}^{A,b} - \delta U_{l,com}^{A,b} - \delta U_{v}^{A,b},
\]

\[
U_{l}^{A,b} = \delta V_{l}^{A,b} - V_{l}^{A} + \sin(\psi^b)(U_{r}^{gust} + \delta U_{r,com}^{gust} - \delta U_{v}^{A,b}) - \cos(\psi^b)(U_{r}^{gust} + \delta U_{r,com}^{gust} - \delta U_{v}^{A,b}),
\]

\[
\alpha^{A,b} = \arctan \left( \frac{U_{n}^{A,b}}{U_{l}^{A,b}} \right) - \phi^{A,b}.
\]

**Step 4** Compute normal $U_{n}^{A,b}$ and lead-wise $U_{l}^{A,b}$ effective wind speeds and angle of attacks.

**Step 5** Compute normal and lead-wise forces and pitch-wise torques per blade element.

\[
\begin{align*}
\delta q_{n}^{A,b} &= \frac{1}{2} \partial \alpha_{A} \left( C_1(\alpha^{A,b}) U_{n}^{A,b} + C_0(\alpha^{A,b}) U_{l}^{A,b} \right) \left[ \sqrt{(U_{n}^{A,b})^2 + (U_{l}^{A,b})^2} \right]^2 - q_{n}^{A,b}, \\
\delta q_{l}^{A,b} &= \frac{1}{2} \partial \alpha_{A} \left( C_0(\alpha^{A,b}) U_{n}^{A,b} - C_1(\alpha^{A,b}) U_{l}^{A,b} \right) \left[ \sqrt{(U_{n}^{A,b})^2 + (U_{l}^{A,b})^2} \right]^2 - q_{l}^{A,b}, \\
\delta q_{w}^{A,b} &= \frac{1}{2} \partial \alpha_{A} \left[ C_3(\alpha^{A,b}) \left[ \left( U_{n}^{A,b} \right)^2 + (U_{l}^{A,b})^2 \right] \right] - q_{w}^{A,b}.
\end{align*}
\]

**Step 6** Update dynamic term on axial induction speed, to be used in next time instant, $\delta U_{i}^{n}$, using the ECN differential equation model.

### 3.4. Conventional controller

The conventional controller is typical and contains two loops: pitch control for generator speed regulation (active above-rated only) and generator torque control for power regulation (according to optimal LQN-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 (fourth-order inverse Chebyshev type II filter with cutoff frequency of (3P + 0.8) rad s$^{-1}$ and 20 dB reduction), band-stop filter around the first turbine towerward frequency $f_{a}$ (second-order elliptic filter with stop-band [0.85$f_{a}$, 1.1$f_{a}$] rad s$^{-1}$, 30 dB reduction and 1 dB ripple) and a band-stop filter at the first collective lead-frequency $f_{b}$ (fourth-order elliptic filter with stop-band [0.8$f_{l}$, 1.05$f_{l}$] rad s$^{-1}$, 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 paper, an extreme rising wind gust with simultaneous wind direction change is simulated. These have been chosen as specified in IEC 61400-1 as ‘extreme coherent gust with direction change (ECD)’: 15 m s$^{-1}$ rising wind gust (on top of the mean wind $\bar{U} = 15$ m s$^{-1}$ and the additional blade-effective turbulence) in conjunction with a direction change of $725/180^\circ$). A simulation of the complete turbine model with the described extreme event occurring 5 s after the beginning of the simulation, is shown in Figure 3. On the top subplot of the figure, the rotor speed $\Omega_{r}$ [the fluctuating (black) curve], together with its filtered version $\Omega^{A}_{r}$ [the smoother (green) curve] is given. The rated speed $\Omega_{r}$, 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). The supervisory system is not modeled in the simulation, so the turbine is not stopped after the rotor speed exceeds the overspeed limit around $t = 9$ s. 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 ($\approx \Omega + 1$ 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 Ip loads, resulting from the oblique infl ow, 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 paper was to develop algorithm for EEC that: (i) is capable of preventing rotor overspeed, when possible; and (ii) 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 $\beta$ by means of a non-linear 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 detail.

4.1. Simplified model

The algorithm for EG&DR utilizes an EKF for the estimation of a so-called augmented state $x'$, 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 $\beta$). 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 ‘Turbine Simulation Model’. This simplified model also consists of an interconnection of an SDM and ADM blocks, although their complexity is somewhat simplified as described below:

4.1.1. 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 lead-wise blade element velocity because of rotor rotation, which is, of course, not neglected). This boils down to setting $\delta V_{\infty b} = \frac{V_{\infty L b}}{\Omega} (\Omega - \bar{\Omega})$ and $\delta V_{\infty b}^h$ in ‘ADM’. Furthermore, the blade element pitch angle variations are assumed to be constant over the blade, i.e. $\delta \phi^h = \delta \phi$, and are assumed measured at the blade roots. The third simplification is that equilibrium wake is considered, being equivalent to setting $\delta U^e = 0$ (and skipping step 6 in the algorithm of ‘ADM’). 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.
4.1.2. 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, but only the 10 measured outputs (i.e. $\Omega$, $\delta \phi$, $M^{u}_{w}$, and $M^{m}_{w}$).

4.1.3. $T_{s}$

The model reduction, mentioned earlier, is performed after re-sampling the SDM model to $T_{s}$ (the sampling time SDM for turbine simulation is $T_{s} = 0.005$ s).

Define the Coleman transformation $T_{a}(\cdot)$ (modulation) and inverse Coleman transformation $T_{d}(\cdot)$ (demodulation).

$$T_{a}(\psi) = \frac{1}{3} \begin{bmatrix} 2 \sin(\psi_{1}) & 2 \sin(\psi_{2}) & 2 \sin(\psi_{3}) \\ 2 \cos(\psi_{1}) & 2 \cos(\psi_{2}) & 2 \cos(\psi_{3}) \end{bmatrix},$$

$$T_{d}(\psi) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & \sin(\psi_{1}) & \cos(\psi_{1}) \end{bmatrix} = T_{d}(\psi).$$

The map $T_{a}$ is used to transform variables, defined in the rotating reference frame, to the non-rotating reference frame (e.g. $M^{u}_{w} = T_{a}(\psi)M_{w}$), while $T_{d}$ is used for the inverse operation.

Using this notation, the simplified model can be compactly described in the following state-space form

$$\begin{align*}
\dot{x}_{k+1} &= Ax_{k} + B_{f}[T_{a}(\psi_{k}) @ 1] \delta \phi_{k} + B_{d}[I @ T_{d}(\psi_{k})] f_{ADM}(\delta \Omega_{k}, \delta \phi, u_{k}, \beta_{k}) \\
\delta M_{k} &= [I @ T_{a}(\psi_{k})] [C_{x} + D_{f}[T_{a}(\psi_{k}) @ 1] \delta \phi_{k} + D_{d}[I @ T_{d}(\psi_{k})] f_{ADM}(\delta \Omega_{k}, \delta \phi, u_{k}, \beta_{k})] \\
\delta \Omega_{k} &= C_{x} x_{k} + D_{f}[T_{a}(\psi_{k}) @ 1] \delta \phi_{k} + D_{d}[I @ T_{d}(\psi_{k})] f_{ADM}(\delta \Omega_{k}, \delta \phi, u_{k}, \beta_{k}) \\
\delta \phi_{k} &= T_{a}(\psi_{k}) (C_{x} x_{k} + D_{f}[T_{a}(\psi_{k}) @ 1] \delta \phi_{k})
\end{align*}$$

where the rotor azimuth $\psi_{k}$ is viewed as known time-varying parameter since $\psi_{k}$ is re-sampled in $f_{ADM}(\delta \Omega_{k}, \delta \phi, u_{k}, \beta_{k})$, but depends only on the rotor speed $\Omega_{k}$ up to time instant $(k - 1)$, but not on $\Omega_{k}$ (and, hence, is not a function of the current state).

The goal was 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 $\beta_{k}$.

4.2. Augmented-state EKF

For the purpose of EG&DR, the unknown inputs $u_{k}$ and $\beta_{k}$ in model (6) 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 \( \beta_k \) are computed. Although blade-effective wind turbulence models do exist,\(^{13}\) 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\(^{14}\) and the references therein). The basic idea behind this approach is to model the unknown input using a random walk model

\[
\begin{bmatrix}
u_{k+1} \\
\beta_{k+1}
\end{bmatrix} = \begin{bmatrix}
u_k \\
\beta_k
\end{bmatrix} + \mathbf{r}_k
\]

(7)

where \( \mathbf{r}_k \) is a zero-mean white Gaussian process with covariance matrix \( \mathbf{R}_c \). Usually, the covariance matrix \( \mathbf{R}_c \) of the noise term \( \mathbf{r}_k \) 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 \( \mathbf{R}_c \) which however results in less smooth (i.e. more noisy) estimates, and vice versa.

Basically, the model (7) 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 (7), the state \( x \) of the system (6) is augmented with the unknown inputs, resulting in the following augmented-state model

\[
\begin{bmatrix}
u_{k+1} \\
\beta_{k+1}
\end{bmatrix} = \begin{bmatrix}
u_k \\
\beta_k
\end{bmatrix} + \mathbf{r}_k
\]

\[
\mathbf{M}_k = \begin{bmatrix} A_k & B_k \end{bmatrix} \mathbf{x}_k + \mathbf{w}_k
\]

Step 3 Compute \( C_k = \partial g(x^e, \psi_k) / \partial x^e \bigg|_{x_k=x^e_{k-1}, \psi_k} \)

Step 4 Measurement update:

\[
\begin{aligned}
K_k &= P_{k-1} C_k (C_k P_{k-1} C_k^T + \mathbf{R}_c)^{-1} \\
\hat{x}_k &= \hat{x}_{k-1} + K_k (\mathbf{M}_k - g(x^e_{k-1}, \psi_k) - \hat{D}_k \delta v_k) \\
P_{k} &= (I-K_k C_k) P_{k-1}
\end{aligned}
\]

Remark 4.1 The EKF requires the partial derivatives of the non-linear functions with respect to the state variables. These can be analytically computed (Kanev and van Engelen,\(^{11}\) App. 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.\(^\text{16,17}\) The author’s experience, however, is that for the model described here it often runs into numerical problems because of the output covariance matrix becoming numerically singular.

### 4.3. CUSUM Test for Extreme Event Detection

The EKF, discussed earlier, estimates the turbine structural model state \( x \), together with the blade-effective wind speeds \( u \) and the oblique inflow angle \( \beta \), contained in the augmented state \( x^e \). Under normal conditions, \( u \) and \( \beta \) will be stochastic signals with zero mean value, while under
extreme conditions their mean values will undergo a change. In order that appropriate EEC 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  and  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 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:

4.3.1. Algorithm 4.2 (CUSUM test)

Initialization Choose integers  (moving window length),  (insensitivity parameter),  (threshold) and set  = 0 (vector with initial wind speed estimates),  = 0.

Update Compute

Detection If  ( || > ), set  = 1, else set  = 0.

The signal  ∈  is 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  starts increasing,  will also increase until  converges, at which point  <  and  will start decreasing to zero again. In this way, an easy detection mechanism would be to put a threshold  on the sum of the elements of the vector  so that an extreme event flag is raised (  = 1) whenever  || > , where  || denotes the vector 1-norm. Once  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  can be pulled down by either the CUSUM test algorithm above (i.e. when  || ≤ ), 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. EEC

This section develops an algorithm for EEC that consists of two parts: (i) collective feedforward pitch control for preventing rotor overspeed; and (ii) IPC 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 in Figure 3, the conventional PI pitch controller is incapable to keep the rotor speed within its limits under extreme wind gusts. The reason for that is that: (i) it reacts on the filtered rotor speed  which is delayed by about 1 s with respect to the true speed  and (ii) 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 ‘CUSUM Test for Extreme Event Detection’, the EEC starts pitching the blades to feather with the maximally allowable pitch speed under extreme conditions  This results in fast reduction of the rotor speed, but has as a side effect a very large tower base fore-aft moment caused by the large reduction of the rotor thrust force. In order to limit the tower base moment, after some time  (about 1 s) the pitching speed is reduced to the maximum pitch speed under normal conditions,  .

The conventional generator torque control at above-rated conditions was designed to achieve constant power, equal to the rated power (see ‘Conventional Controller’). This implies a negative generator torque sensitivity to rotor speed variation, i.e.  . This has a destabilizing effect on the rotor speed, which is stabilized by the pitch control algorithm. However, because of 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  . 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  is pulled down to zero by CUSUM algorithm in ‘CUSUM Test for Extreme Event Detection’, or whenever the pitch angle  gets ‘close’ to a reference pitch angle  dependent on the estimated axial wind speed  .

\[
\dot{\theta}_{\text{ext,ax}}^g = \cos(\theta_b) \cos(\theta_b + \hat{\beta}_k) \left( \bar{T}_g + \frac{1}{3} \sum_{i=1}^{3} \hat{a}_{b,k} \right) \quad (9)
\]

More specifically,  is defined as the collective pitch angle that, for axial wind speed  , rated rotor speed  and rated generator torque  , achieves azimuth-averaged static aerodynamic torque  . For a given  ,  is computed by solving the following non-linear optimization problem

\[
\theta_{\text{ext,ax}}^g = \arg\min_{\theta} \left| \bar{T}_g(\Omega, \theta, U_{\text{ext,ax}}^g) - \bar{T}_g \right|
\]
The function $\theta_{\text{ref},k}(\hat{U}_{\text{ax}}^{\text{ext}})$ is numerically computed offline and stored for different values of $\hat{U}_{\text{ax}}^{\text{ext}}$. Simple linear interpolation is then performed online.

To avoid unnecessary on/off switchings of the EEC because of fluctuations in $\theta_{\text{ref},k}(\hat{U}_{\text{ax}}^{\text{ext}})$, hysteresis is introduced: the EEC will switch on only when the extreme event flag gets raised (i.e. $f_{\text{ec,k}} = 1$ and $f_{\text{de,k}} = 0$), and the current collective pitch angle is at least $\Delta \theta_{\text{ax}}^{\text{ref}}$ (e.g. $5^\circ$) below the reference pitch angle. The extreme event flag gets pulled down to zero ($f_{\text{ec,k}} = 0$), 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 $\theta_{\text{ref},k}(\hat{U}_{\text{ax}}^{\text{ext}})$ and the true current collective pitch angle drops below $\Delta \theta_{\text{ax}}^{\text{ref}}$ (e.g. $4^\circ$), meaning that no further EEC action is needed. The rotor speed limitation algorithm can be summarized as follows.

### 5.1. Algorithm 5.1 (Collective EEC)

#### Initialization

Select $\Delta \theta_{\text{ax}}^{\text{ref}}$, $\Delta \theta_{\text{ax}}^{\text{ref}} < \Delta \theta_{\text{ax}}$, $t_{\text{rec}} = 0$.

**Step 1** Use the current EKF estimates $\hat{u}_k$ and $\hat{b}_k$ to compute $\hat{U}_{\text{ax},k}^{\text{ext}}$ using equation (9).

**Step 2** Run CUSUM test in algorithm 4.2. If $f_{\text{ec,k}} = 0$, then set $t_{\text{rec}} = 0$ and go to step 5.

**Step 3** Compute $\Delta \theta_{\text{ax},k} = \theta_{\text{ref},k}(\hat{U}_{\text{ax}}^{\text{ext}}) - \frac{1}{2} \Sigma_{i=1}^k \phi_i$.

**Step 4** If $(f_{\text{ec,k}} = 1$ and $\Delta \theta_{\text{ax},k} \geq \Delta \theta_{\text{ax}}^{\text{ref}})$ or $(f_{\text{ec,k}} = 0$ and $\Delta \theta_{\text{ax},k} < \Delta \theta_{\text{ax}}^{\text{ref}})$ then

\[
\text{switch conventional control off}
\]

\[
t_{\text{rec}} \leftarrow t_{\text{rec}} + T_{\text{ut}},
\]

\[
\theta_k = \begin{cases} 
\theta_{k-1} + \theta_{\text{ax,ext}} T_{\text{ut}} & \text{if } t_{\text{rec}} \leq \Delta t_{\text{rec}}, \\
\theta_{k-1} + \theta_{\text{ax}} T_{\text{ut}} & \text{otherwise}.
\end{cases}
\]

\[
T_{\text{ut},k} = \hat{T}_k.
\]

else

\[
\text{$(f_{\text{ec,k}} = 1 \text{ and } f_{\text{de,k}} = 0)$, then}
\]

\[t_{\text{rec}} = 0,
\]

\[f_{\text{de,k}} = 1.
\]

**Step 5** If $f_{\text{ec,k}} = 1$ and $f_{\text{de,k}} = 0$, then

\[
\text{re-initialize conventional pitch control}
\]

\[\text{switch on conventional control}.
\]

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 re-initialized before being switched on. This can be achieved by considering an interval of $N$ time steps back, $[k - N, k - 1]$, and choosing the state of the conventional controller at time $(k - N)$ in such a way that, if the conventional controller was active in the interval $[k - N, k - 1]$, it would have produced a control signal that matches the true control signal observed in this interval. This is described in more detail in Kanev and van Engelen (App. B).

### 5.2. Blade load reduction

As mentioned in the beginning of ‘EEC’, besides rotor overspeed prevention, an important issue under extreme wind gusts with direction change is the reduction of blade loads. A yawed wind infl ow 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 non-linear model (4) is linearized at a given operating point, resulting in the following linear model in Coleman domain

\[
\mathcal{T}:
\begin{align*}
\dot{x}_{k+1} &= \bar{A}x_k + \bar{B} \theta_k + \bar{B}_u u_k^0, \\
M_k^0 &= \bar{C} \theta_k + \bar{D}_u u_k^0,
\end{align*}
\]

where the signals $v_k^0$, $\theta_k^0$ and $M_k^0$ contain the tilt and yaw-oriented components of the multiblade blade effective wind speed vector $u_k^0$, blade pitch angles $\theta_k^0$ and flap-wise blade root bending moments $M_k^0$, respectively. The considered extreme event in this report (gust with direction change) can be modeled by a non-zero constant tilting (i.e. first) component in $\theta_k$. 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 $\theta_k$ also barely affect the rotor speed dynamics and need not be taken into consideration in the conventional rotor speed control design.

The goal here was to design a stabilizing controller that uses the rotor moments $M_k^0$ as inputs and computes the control actions $\theta_k^0$ 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 the first tower frequency $f_{\text{tower}}$. 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 $u_k^0$ to some suitable chosen weighted versions of the rotor moments and control action. More specifically, Figure 4 provides a 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 $\mathcal{K}_{\text{ipc}}$. 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 in 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 unmodeled dynamics, an $H_\infty$ 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 \cite{12} can be 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 paper, 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 $\mathcal{K}_{\text{ipc}}$ is designed by minimizing the $H_\infty$ norm of the closed-loop transfer from the external inputs $u^I_k$ 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 in Figure 5. For producing the left subplot in Figure 5, the weighting function for the control signals has been chosen as

$$W_u(z) = 10[F_{hp}(z) + F_{3p}(z)F_{6p}(z)F_{tor}(z) - 2]I_2$$  \hspace{1cm} (10)

where $F_{hp}(z)$ is a second-order inverse Chebyshev high-pass filters (frequency $f_{hp} = 4P$, reduction 20 dB, ripple 1 dB), and $F_{3p}(z)$, $F_{6p}(z)$ and $F_{tor}(z)$ 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 $[0.9, 1.1]f_{tor}$, respectively. All filters have been scaled to achieve unity DC gain, so that $W_u$ computed via (10) has a DC gain of zero. The so-selected weighting function $W_u$ punishes control activity at frequencies $f_{tor}$, $f_{3p}$, $f_{6p}$ 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 s$^{-1}$, lag frequency of $5$ rad s$^{-1}$ and DC gain of $20$. Notice that $W_M$ acts on the integrated rotor moments. Translating this to the original the rotor moments $M_0$, this results in some additional weighting of the frequency band $[1, 5]$ rad s$^{-1}$.

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

$$T^a(z) = z_k = \begin{bmatrix} 0 & W_a(q^{-1}) \\ T^u & W_a(q^{-1}) T(q^{-1}) \end{bmatrix} \begin{bmatrix} \theta_i^a \\ \theta_i^a \\ \theta_i^a \\ \theta_i^a \\ \theta_i^a \end{bmatrix}
$$

The $H_\infty$ optimal controller for $T^a(z)$ is computed via the following optimization problem

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

where $\mathcal{F}(T^a(z), K(z))$ denotes the closed-loop system, $\| \cdot \|_{\infty}$ denotes the $H_\infty$ system norm and wherein the optimization is defined over all controllers $K(z)$ that have the same number of states as the augmented model $T^a(z)$. For more details on modern robust control design, the reader is referred to Zhou and Doyle.\textsuperscript{19} 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.

$$\mathcal{K}^i_{IPC} = \mathcal{K}_{IPC} \begin{bmatrix} T^u \\ \frac{z}{z-1} \end{bmatrix}
$$

### 6. SIMULATION

The performance of the complete algorithm, including extreme event recognition and control, is demonstrated on simulation data, obtained with the non-linear test turbine model described in ‘Turbine Simulation Model’. The model represents a three-bladed HAWT with rated power of $2.5$ MW, rotor radius of $R = 40$ m and rated rotor speed of $\Omega = 1.85$ rad s$^{-1}$. 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$^{-1}$ [with $\bar{\phi}_b = -5.138^\circ$ (mainly caused by tilted rotor) and $\bar{\phi}_y = 0.01^\circ$] and blade pitch angles of $\phi^a = 7.24^\circ$. The values selected for the tuning parameters of the EG&DR and EEC schemes are given in Table I. In order to evalu-

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Variable</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EKF</td>
<td>$n$</td>
<td>20</td>
<td>State dimension</td>
</tr>
<tr>
<td></td>
<td>$\hat{x}_0$</td>
<td>0</td>
<td>Initial state estimate</td>
</tr>
<tr>
<td></td>
<td>$P_0$</td>
<td>$\begin{bmatrix} 10^{-6}I_{n3} &amp; 0 \ 0 &amp; 10^{-5} \end{bmatrix}$</td>
<td>Initial state covariance matrix</td>
</tr>
<tr>
<td></td>
<td>$R_1$</td>
<td>$\begin{bmatrix} 10^{-2}I_2 \ 10^{-4} \end{bmatrix}$</td>
<td>Process noise covariance matrix</td>
</tr>
<tr>
<td></td>
<td>$R_a$</td>
<td>$\begin{bmatrix} 10^3I_2 \ 10^2I_2 \end{bmatrix}$</td>
<td>Measurement noise covariance matrix</td>
</tr>
<tr>
<td>CUSUM</td>
<td>$k_t$</td>
<td>25</td>
<td>Moving window length</td>
</tr>
<tr>
<td></td>
<td>$\nu$</td>
<td>1</td>
<td>Insensitivity parameter</td>
</tr>
<tr>
<td></td>
<td>$h$</td>
<td>100</td>
<td>Threshold</td>
</tr>
<tr>
<td>EEC</td>
<td>$\theta_{max,ext}$</td>
<td>$10^5$ s$^{-1}$</td>
<td>Max pitch speed under extreme event</td>
</tr>
<tr>
<td></td>
<td>$\theta_{max}$</td>
<td>$4^\circ$ s$^{-1}$</td>
<td>Max pitch speed under normal conditions</td>
</tr>
<tr>
<td></td>
<td>$\Delta\theta_{eti}$</td>
<td>$5^\circ$</td>
<td>EEC activation zone</td>
</tr>
<tr>
<td></td>
<td>$\Delta\theta_{eti}$</td>
<td>$4^\circ$</td>
<td>EEC deactivation zone</td>
</tr>
<tr>
<td></td>
<td>$F_{3p}$</td>
<td>$z^4 - 3.973z^3 + 5.948z^2 - 3.977z + 1.002$</td>
<td>3p Band pass filter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$z^4 - 3.953z^3 + 5.883z^2 - 3.908z + 0.9774$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$F_{6p}$</td>
<td>1</td>
<td>6p Band pass filter</td>
</tr>
<tr>
<td></td>
<td>$F_{4\omega}(z)$</td>
<td>1</td>
<td>Tower frequency band pass filter</td>
</tr>
<tr>
<td></td>
<td>$F_{act}(\theta)$</td>
<td>$\frac{10z^2 - 19.48z + 9.57}{z^2 - 1.554z + 0.6415}$</td>
<td>Control signal weighting filter</td>
</tr>
<tr>
<td></td>
<td>$F_{act}(\theta)$</td>
<td>$100z - 98.1$</td>
<td>Integrated rotor moments lead-lag filter</td>
</tr>
</tbody>
</table>

---

DOI: 10.1002/we
M. Kanev and T. van Engelen

Table II. Simulated wind gust cases.

<table>
<thead>
<tr>
<th>Case</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>(V_{\text{gust}}) (m s(^{-1}))</td>
<td>15</td>
<td>15</td>
<td>3</td>
</tr>
<tr>
<td>(\beta_{\text{gust}}) (deg)</td>
<td>48</td>
<td>30</td>
<td>-3</td>
</tr>
</tbody>
</table>

* Note that the tilt and yaw components (\(u^t\)) of the multiblade 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 2). 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 \(u^t\) 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.

Different simulations are run. 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 presented in the figures below, only the first 20 s are plotted. The (extreme) events occur 5 s from the beginning of each simulation. For the power spectra plots later on, the time series from the 10th second to the end of the simulations.
are used, so that only the data after the event occurrence (and after the transients have died out) are 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 not get activated by the EG&DR scheme as the event is not recognized as major.

6.1. Evaluation of the EG&DR

The performance of the EG&DR scheme is determined by the accuracy of the estimates of the EKF. To evaluate that, we will compare the simulated blade-effective wind speeds

![Figure 7. Turbine simulation under case 1 (extreme 15 m s⁻¹ rising gust and 48 deg direction change at t = 5 s) without extreme event control (EEC) (left) and with EEC (right).]
\( u_b \) and the simulated wind direction change angle \( \beta \) to their estimates, computed by the EKF.

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, although these are not reported here for the sake of brevity. The remaining three subplots in Figure 6 depict the simulated oblique inflow angle \( \beta \) (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 since the estimates do not differ more than about \( \pm 3 \) degrees from the simulated values.

### 6.2. Evaluation of the EEC

As discussed in ‘Problem Formulation’, the purpose of the EEC algorithm was 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 \( \Omega \), the blade pitch angles \( \phi \) and the blade root out-of-plane bending moments \( M^b_z \) are next investigated under the 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 \( \Omega \) getting much above its limit. This is because of the conventional controller remaining in partial load regime until the filtered rotor speed \( \Omega_f \) (dashed green line) exceeds the rated speed \( \Omega_r \) by 1 rpm, at which point the true speed \( \Omega \) is already too large. The EEC algorithm, on the other hand, detects the gust at an early stage (at time 6.125 s) 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 in Figure 7 during the second half of the simulation (where the IPC is active). The achieved blade load reduction can also be appreciated by observing the left subplot in Figure 8 that depicts the spectra of the blade root out-of-plane bending moment variations \( \delta M^b_z \) in the cases without (solid red curve) and with (dashed black curve) EEC. The simulation results under case 2 are depicted in Figure 9. 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 9) 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 9), 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 9, as well as from the power spectra in the right-hand side subplot of Figure 8.

Finally, case 3 is simulated only once, i.e. with the EEC algorithm on, although it does not get activated by the EG&DR 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 simulation. The rotor speed \( \Omega \), the blade pitch angles \( \phi \) and the blade root out-of-plane bending moments \( M^b_z \) 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

![Figure 8](https://example.com/f8.png)

**Figure 8.** Power spectral density of blade root flap-wise bending moments \( M^b_z \) for case 1 (left) and case 2 (right), without extreme event control (EEC) (solid curves) and with EEC (dashed curves).
$M^b_z$ after the event occurrence remain comparable to those before the gust.

7. CONCLUSION

Extreme wind gust with direction change can cause turbine shutdown because of 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 paper. To avoid this, an algorithm for extreme event recognition and control is developed that uses: (i) an EKF to estimate the turbine states together with the blade-effective wind speeds.

Figure 9. Turbine simulation under case 2 (extreme 15 m s\(^{-1}\) rising gust and 30 deg direction change at $t = 5$ s) without extreme event control (EEC) (left) and with EEC (right).
Figure 10. Turbine simulation under case 3 (3 m s$^{-1}$ rising gust and $-3 \text{ deg}$ direction change at $t = 5 \text{ s}$). Because of the mild gust condition, the extreme event control does not get activated.
and oblique wind inflow angle; (ii) a CUSUM algorithm to detect changes in the mean of the estimated wind signals; (iii) a fast feedforward collective pitch control algorithm for rotor overspeed prevention; and (iv) a feedback IPC algorithm for 1p blade load reduction. The complete algorithm is demonstrated in different non-linear simulations with a 40th order (linearized) structural dynamics model, obtained with the software TURBU, detailed non-linear BEM aerodynamics and realistic blade-effective wind speed signals.

REFERENCES