

Article

Implementation and Validation of an Advanced Wind Energy Controller in Aero-Servo-Elastic Simulations Using the Lifting Line Free Vortex Wake Model

Sebastian Perez-Becker *, David Marten, Christian Navid Nayeri and Christian Oliver Paschereit

Chair of Fluid Dynamics, Hermann Föttinger Institute, Technische Universität Berlin, Müller-Breslau-Str. 8, 10623 Berlin, Germany; david.marten@tu-berlin.de (D.M.); christian.nayeri@tu-berlin.de (C.N.N.); oliver.paschereit@tu-berlin.de (C.O.P.)

* Correspondence: s.perez-becker@tu-berlin.de

Abstract: Accurate and reproducible aeroelastic load calculations are indispensable for designing modern multi-MW wind turbines. They are also essential for assessing the load reduction capabilities of advanced wind turbine control strategies. In this paper, we contribute to this topic by introducing the TUB Controller, an advanced open-source wind turbine controller capable of performing full load calculations. It is compatible with the aeroelastic software QBlade, which features a lifting line free vortex wake aerodynamic model. The paper describes in detail the controller and includes a validation study against an established open-source controller from the literature. Both controllers show comparable performance with our chosen metrics. Furthermore, we analyze the advanced load reduction capabilities of the individual pitch control strategy included in the TUB Controller. Turbulent wind simulations with the DTU 10 MW Reference Wind Turbine featuring the individual pitch control strategy show a decrease in the out-of-plane and torsional blade root bending moment fatigue loads of 14% and 9.4% respectively compared to a baseline controller.

Keywords: wind energy; wind turbine control; load mitigation; individual pitch control; lifting line free vortex wake; vortex methods



Citation: Perez-Becker, S.; Marten, D.; Nayeri, C.N.; Paschereit, C.O. Implementation and Validation of an Advanced Wind Energy Controller in Aero-Servo-Elastic Simulations Using the Lifting Line Free Vortex Wake Model. *Energies* **2021**, *14*, 783. <https://doi.org/10.3390/en14030783>

Academic Editor: Robert Castilla
Received: 6 January 2021
Accepted: 28 January 2021
Published: 2 February 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Over the past years, wind turbines have increased significantly in size in an effort to reduce the cost of wind energy and make it a competitive source of energy. So far, this has been a successful approach. An example of this success can be seen in the increased installation numbers of new wind turbines in Germany [1]. Yet increasing the turbine size also has its drawbacks. A larger rotor will be subjected to higher aerodynamic and gravitational loads. For example, the aerodynamic bending moments and the moments due to the self-weight of the blade scale as the third and fourth power of the rotor diameter respectively ([2], (pp. 97–123)). As a consequence, the structure of the turbine components such as rotor blades has to be stiffer, which requires more or stronger material. This leads to an increase in the cost of energy. A way of counteracting the load increase seen in larger wind turbines is through the use of advanced wind turbine controllers. This allows for less material use in the design of the different components and hence results in a decrease of the cost of energy.

The most common actuator used for load alleviation is the blade pitch actuator. This comes from it being already used in the power regulation strategy of modern turbines. Many advanced load alleviation strategies using pitch actuators have been proposed. One of the best-known ones is the Individual Pitch Control (IPC) strategy [3], which commonly relies on the out-of-plane Blade Root Bending Moments (BRBM) of the individual blades as input signals. It has been used in combination with different input sensors—such as inflow sensors [4,5]—and also in combination with different actuators—such as active

trailing edge flaps [6–9]. Other studies have studied advanced model-based [10] or adaptive [11] controllers for load alleviation. A controller that uses neural networks as part of its architecture has also been recently proposed in [12].

Oftentimes, the results of these studies are difficult to compare and reproduce. This is partly because many research groups use self-developed baseline controller strategies as a basis for load reduction comparative studies. The source code of these controllers is rarely available. Alternatively, studies that use the NREL 5 MW Reference Wind Turbine (RWT) [13] also use the baseline controller available in the model definition. This is an older controller—based on [14,15]—that offers limited functionality and is inconvenient to use with other turbine models. This is because the controller parameters are hard-coded in the source file and the controller has to be recompiled every time the parameters change.

As part of the UpWind Project, a more advanced baseline controller was developed which allows for better power tracking and smoother transitions between the operating regions. These features lead to better energy capture and to reduced loading compared to the NREL 5 MW controller. Many of the aforementioned features are also implemented in the Basic DTU Wind Energy Controller [16].

In recent years, the wind energy community has started to address the problem of reproducible controller results by introducing several modern, open-source reference wind turbine controllers. Mulders and van Wingerden publish the Delft Research Controller (DRC) [17], which is expanded by Abbas et al. into NREL's Reference Open-Source Controller (ROSCO) [18]. Meng et al. also extend the Basic DTU Wind Controller to form the DTU Wind Energy Controller (DTUWEC), which includes advanced industrial features [19].

All of the cited reference controllers use classical controller architectures such as PID controllers. As discussed in ([20], (pp. 506–518)), there are several reasons for this choice. An important one is that with classical controller architectures, controller stability can be guaranteed. This is of utmost importance since modern large wind turbines are expected to operate reliably under all circumstances with minimum supervision. Another reason is that individual features can be added to the controller without the need to re-calibrate the whole controller. Also, the integration of a supervisory controller is much more straightforward when classical controller architectures are used. Other control techniques such as neural networks are powerful techniques that could potentially be used to solve specific control objectives. They are therefore well suited to be part of advanced controller features. They are however less appropriate as candidates for reference baseline controllers. Because of their black-box model nature, it is difficult to guarantee their stability, add specific features or integrate them with supervisory controllers.

Having a reference controller is one aspect of accurate and comparable load reduction estimation. In order to have a complete picture of the design loads of different wind turbine components, we need to simulate the wind turbine in realistic aeroelastic scenarios that often include controller faults or other unforeseen events. Current industry standards prescribe a large number of aeroelastic simulations of the complete turbine [21]. These are grouped together into Design Load Case (DLC) groups, each considering a particular scenario in the wind turbine's design life. Most of the aforementioned studies only include a selection of DLC groups, such as the power production DLC group. While being a good approximation for certain component's loads, using a selection of DLCs will not give a complete picture of the load reduction capabilities of the different turbine controllers. This is in part because many of the available wind turbine controllers do not feature a supervisory controller. The latter oversees the pitch and torque controller and reacts to unforeseen events, shutting down if threshold values of certain signals are passed in order to ensure the structural integrity of the turbine components. Only a full load calculation according to industry standards will give an accurate estimate of the load reduction capabilities of advanced controller strategies.

Another important aspect for accurate load estimation is the use of appropriate models in the aeroelastic simulations. Current aeroelastic codes mostly rely on the Blade Element

Momentum (BEM) aerodynamic model to calculate aerodynamic loads ([20], (pp. 57–66)). BEM models are attractive because they are computationally inexpensive. Yet in order to capture the more challenging unsteady aerodynamic phenomena present in DLCs, BEM models require a series of engineering corrections. These corrections have been developed and tested so that they work for a wide range of operating conditions. If the wind turbine operates in a condition outside this range, then the BEM corrections could introduce inaccuracies and overestimate the aerodynamic loading on the turbine. Examples of this include turbine operation in extreme yawed conditions or inflow conditions that are inhomogeneous across the rotor. This could arise if the turbine is operating in the partial wake of another turbine, in sheared and/or turbulent inflow due to the Earth's boundary layer or if there is a large difference in the individual pitch angles of the blades (e.g., with pitch actuator faults) [22–25].

Vortex methods such as the Lifting Line Free Vortex Wake (LLFVW) aerodynamic model have a higher order representation of the unsteady aerodynamic phenomena and are capable of modeling these with far fewer assumptions than BEM models [26]. This is because in the LLFVW method the wake is explicitly modeled. In [27] the authors show that there are significant differences in fatigue and extreme loading if the aeroelastic simulations are performed with a BEM aerodynamic model compared to a LLFVW model, with the BEM model predicting increased fatigue loads. If we wish to accurately assess the load reduction potential of advanced control strategies based on individual pitch action, we require an accurate representation of the local aerodynamic effects that occur on each blade. Since the advanced controller action will be individual on each blade, the resulting induction field will be non-homogeneous. This in turn requires a higher-order aerodynamic model such as the LLFVW model to accurately estimate the effect of the resulting aerodynamic loads.

In this study, we address the aforementioned issues by introducing the TUB Controller (TUBCon). It is an open-source reference wind turbine controller with advanced load reduction strategies that features a complete supervisory controller. It can therefore be used to perform a full load calculation so that an accurate load picture of the turbine loads is obtained. Furthermore, this controller is fully compatible with the aeroelastic software QBlade [28]—which features the LLFVW aerodynamic model. In combination, QBlade and TUBCon can be used to accurately calculate wind turbine loads and also evaluate the performance of advanced load reduction strategies. We give an example of this by analyzing the load reduction capabilities of the well-known IPC strategy in power production with turbulent wind conditions. This paper is structured as follows: In Section 3 we present and fully describe TUBCon, including its advanced load reduction strategy and supervisory controller. In Section 4, we validate TUBCon in steady and turbulent wind conditions by comparing its performance to an established wind turbine controller from the literature. In Section 5 we analyze TUBCon's advanced load reduction capabilities in turbulent wind conditions by simulating a reference wind turbine using the baseline and IPC variants of TUBCon. Conclusions are drawn in Section 6.

2. Methods

We chose the DTU 10 MW RWT as the turbine model for this study as it is representative of the new generation of wind turbines and has been used in several research studies. The complete description of the turbine can be found in [29].

As explained in [25], large rotors such as the one from the DTU 10MW RWT have similar scales to the scales from turbulent wind found in Earth's boundary layer. As a consequence, the wind field that the turbine rotor sees will be much less homogeneous than the wind field seen by a smaller turbine rotor. BEM codes typically average the induction factor across a blade annulus, thus leading to inaccurate load predictions. In addition, large blade deflections that occur in modern large blades may introduce radial flow, thus violating the assumptions of many BEM models. Both of the aforementioned issues are addressed with higher order aerodynamic models such as the LLFVW model found in TU Berlin's

aeroelastic simulation tool QBlade. We did the testing and simulation of the TUB Controller presented in this study using QBlade.

QBlade

To model the unsteady aerodynamics, QBlade uses the Lifting Line Free Vortex Wake (LLFVW) method [28]. Here, the blade aerodynamic forces are evaluated on a blade element basis using standard airfoil polar data. The wake is modeled with vortex line elements. These are shed at the blade's trailing edge during every time step and then undergo free convection behind the rotor. Vortex methods can model the wake with far less assumptions and engineering corrections compared to BEM methods. Especially when the wind turbine is subject to unsteady inflow or varying blade loads, the LLFVW method increases the accuracy compared to BEM methods [27]. To model the dynamic stall of the blade elements, QBlade uses the ATEFlap unsteady aerodynamic model [30], modified so that it excludes contribution of the wake in the attached flow region [31].

Regarding the structural model, QBlade uses the open-source multi-physics library CHRONO [32]. It features a multi-body representation of the turbine which includes Euler-Bernoulli beam elements in a co-rotational formulation. With it, QBlade is able to accurately simulate blade deflections including the blade torsion, which has a significant influence on the blade loads. A detailed comparison between QBlade and OpenFAST can be found in [27].

3. Description of the TUB Controller

This section describes the TUB Controller (TUBCon). A graphical representation of the controller architecture is shown in Figure 1.

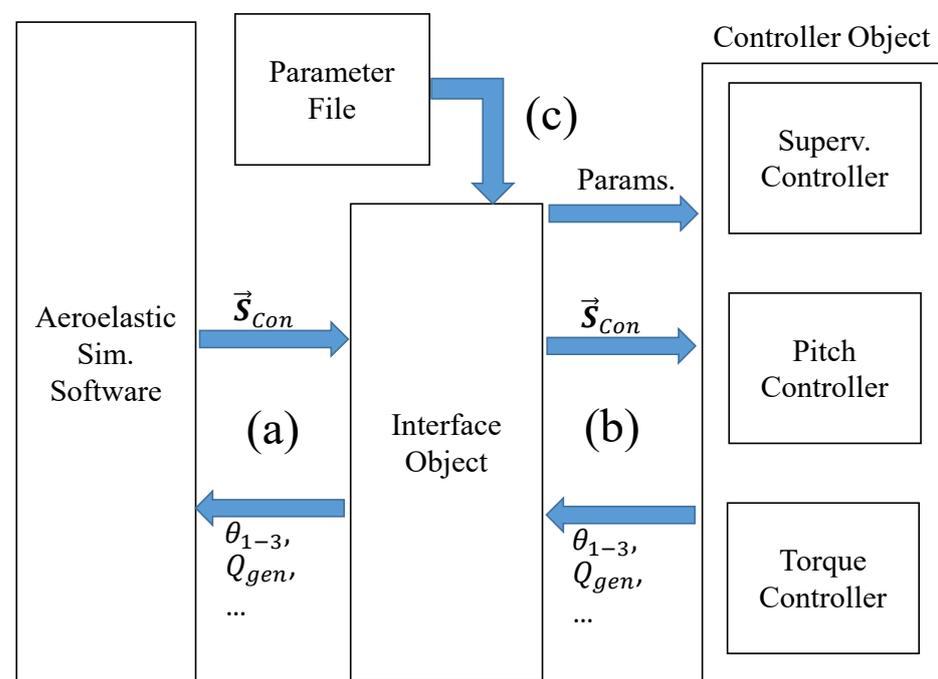


Figure 1. Graphical representation of the controller architecture. The communication with the aeroelastic software is done in two stages using an interface and a controller object. The controller parameters are passed through an external parameter file. \vec{S}_{Con} stands for controller signals.

The controller code is written in C++ in an object-oriented manner. It features a two-stage interaction with the aeroelastic simulation software. The first stage is an interface object that handles the data exchange with the aeroelastic code (Figure 1(a)). The current supported interfaces are the conventional Bladed-Style communication via a swap-array and a modified swap-array communication specially designed for interacting with QBlade.

This makes TUBCon compatible with common aeroelastic simulation codes including OpenFAST, Bladed and QBlade. The modular nature of the interface object makes it easy to extend the compatibility of TUBCon with other aeroelastic codes.

The interface object then communicates with the turbine controller in a second stage Figure 1(b). The individual controller components are designed in Simulink and compiled to a C++ object using the Simulink Coder package. The resulting object has an initialization function, a step function and a termination function. These are called by the main routine at the appropriate simulation times. Such an architecture allows us to rapidly develop and test new controller features that can be incorporated to the compiled controller library. The controller is parameterized externally using an appropriate parameter file (Figure 1(c)). This XML file is read and the controller parameters set as part of the initialization routine.

The controller object is adapted from the Basic DTU Wind Energy Controller [16] and shares many of its capabilities. It features a state-of-the-art pitch and torque controller that enable the turbine to operate reliably under unsteady turbulent wind conditions. Although the description of the torque and collective pitch controller is given in [16], it is also included here for the sake of completeness.

Figure 2 shows a schematic overview of the controller.

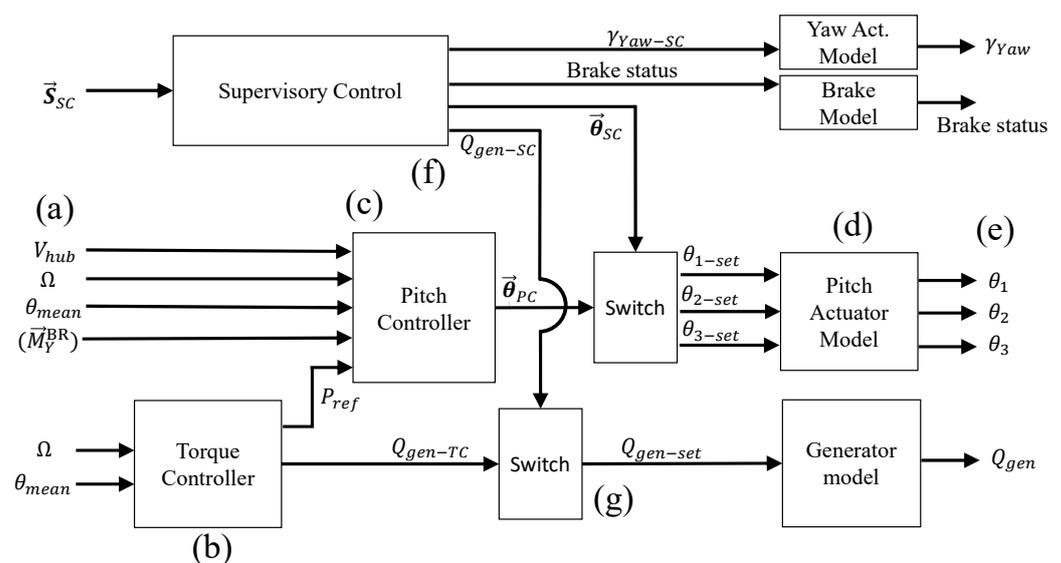


Figure 2. Overview of the controller object. It is comprised of three major modules: the pitch controller, the torque controller and the supervisory control. The latter is able to override the former two with the use of switches. The setpoint signals are passed to the respective actuator model to calculate the actuation signal.

Both the pitch and torque controllers share the rotor speed Ω and the mean pitch angle θ_{mean} as input sensors (Figure 2(a)). The output of the torque controller is the generator set-point torque $Q_{gen-set}$ and the reference power P_{ref} . The pitch controller uses the aforementioned Ω and θ_{mean} as well as P_{ref} and the measured wind speed at hub height V_{hub} to determine the set-point angles of all three blades (θ_{1-set} , θ_{2-set} and θ_{3-set}) (Figure 2(c)). If the IPC strategy is enabled, then the pitch controller also needs the out-of-plane BRBM of all blades (\vec{M}_Y^{BR}). The controller includes actuator models for the pitch and yaw actuators as well as the brakes and generator to model the dynamics of the different actuators (Figure 2(d)). The set-point signal of each controller is passed to the respective actuator model to calculate the actual signals passed back to the aeroelastic code (Figure 2(e)). In the following sections, the individual parts of the controller will be further explained.

Figure 2 also shows the supervisory control. It is included to allow the controller to perform full load calculations. It will be explained in Section 3.3 and is included in Figure 2(f),(g) to show the location where the supervisory control overrides the signals of the individual controllers.

The pitch and torque controllers explained in the following sections operate using the low speed shaft side of the drive train. That is, they assume that the generator speed and torque are identical to the low speed shaft speed and torque. For geared turbines, the gear box ratio G is applied to convert the high speed shaft quantities into low speed shaft quantities and back.

3.1. Torque Controller

The torque controller is divided into two main parts: the partial and the full load regimes. There is also the switching logic to change between both regimes. The input for the controller is the rotor speed Ω and the mean pitch angle θ_{mean} . The latter is used for the switching logic between the two load regimes. The output of the controller is the generator torque Q_{gen} and the reference power P_{ref} for the pitch controller. The reference power is simply the product of the instantaneous generator torque and the rotor speed: $P_{ref}(t) = Q_{gen}(t) \cdot \Omega(t)$.

The variable speed controller is a PID controller with two different rotor speed set-points. The controller is saturated with a maximum and a minimum limit so that Ω is adjusted to follow an optimum tip speed ratio in the partial load regime.

As a first step, the rotor speed Ω is low-pass filtered with a second order low-pass filter to exclude unwanted high frequency dynamics. The function of the low-pass filter is given by Appendix A Equation (A2). The speed error e_{TC} is calculated using the difference of the low-passed rotor speed Ω_{LP} and the current speed set-point Ω_{set} : $e_{TC} = \Omega_{LP} - \Omega_{set}$. The speed set-point is defined as

$$\Omega_{set} = \begin{cases} \Omega_{min} & \text{if } \Omega_{LP} < \frac{\Omega_{rated} + \Omega_{min}}{2} \\ \Omega_{rated} & \text{if } \Omega_{LP} \geq \frac{\Omega_{rated} + \Omega_{min}}{2}, \end{cases} \quad (1)$$

where Ω_{rated} and Ω_{min} represent the rated and minimum rotor speed. The generator torque is computed using the equation

$$Q_{PID}(t) = k_{P-TC} \cdot e_{TC}(t) + k_{I-TC} \cdot \int_0^t e_{TC}(\tau) d\tau + k_{D-TC} \cdot \frac{de_{TC}(t)}{dt}. \quad (2)$$

Here, k_{P-TC} , k_{I-TC} and k_{D-TC} represent the proportional, the integral and the differential gain of the PID controller. Because the values of the generator torque are saturated, each time step, the integral term of Q_{PID} is recalculated as

$$Q_I(t) = k_{I-TC} \cdot \int_0^t e_{TC}(\tau) d\tau = Q_{PID}(t) - k_{P-TC} \cdot e_{TC}(t) - k_{D-TC} \cdot \frac{de_{TC}(t)}{dt}. \quad (3)$$

This helps to avoid windup of Q_I and enables the controller to react quickly if the required torque signal changes from a previously saturated value.

3.1.1. Partial Load Regime

In order for the torque controller to enforce an optimum tip speed ratio in the partial load regime, the generator torque is saturated with upper and lower limits: Q_{max}^P and Q_{min}^P . Between Ω_{min} and Ω_{rated} , the limits are identical and follow the optimal torque speed curve of the turbine:

$$Q_{max}^P = Q_{min}^P = K_{opt} \cdot \Omega_{LP}^2 \quad (4)$$

$$= \frac{\pi \rho R^5 C_{p-opt}}{2 \lambda_{opt}^3} \cdot \Omega_{LP}^2. \quad (5)$$

In the last equation, ρ represents the air density, R the rotor radius and C_{p-opt} the power coefficient at optimum tip speed ratio λ_{opt} .

At two given rotor speed ranges, $[\Omega_{min-a}, \Omega_{min-b}]$ and $[\Omega_{max-a}, \Omega_{max-b}]$, the torque limits open up to allow the PID controller to keep the rotor speed at the required set-point. The switching logic for Q_{min}^P is given by:

$$Q_{min}^P = \begin{cases} \min(K_{opt} \cdot \Omega_{LP}^2 \cdot \sigma(\Omega_{min-a}, \Omega_{min-b}, \Omega_{LP}), K_{opt} \cdot \Omega_{max-a}^2) & \text{if } K_{opt} \cdot \Omega_{max-a}^2 \leq Q_{ref}^F \\ Q_{ref}^F & \text{if } K_{opt} \cdot \Omega_{max-a}^2 > Q_{ref}^F \end{cases} \quad (6)$$

In the above equation, Q_{ref}^F represents the generator torque in full load regime and $\sigma(\cdot)$ represents a smooth switching function between the two limits Ω_{min-a} and Ω_{min-b} . The switch function is given by Equations (A5) and (A6). Complementary to Equation (6), Q_{max}^P follows the switching logic:

$$Q_{max}^P = \max\left((1 - \sigma_{max}) \cdot K_{opt} \cdot \Omega_{LP}^2 + \sigma_{max} \cdot Q_{ref}^F, K_{opt} \cdot \Omega_{min-b}^2\right). \quad (7)$$

The above equation uses the symbol σ_{max} to denote the switch function $\sigma(\Omega_{max-a}, \Omega_{max-b}, \Omega_{LP})$.

Figure 3 shows the torque limits as a function of Ω .

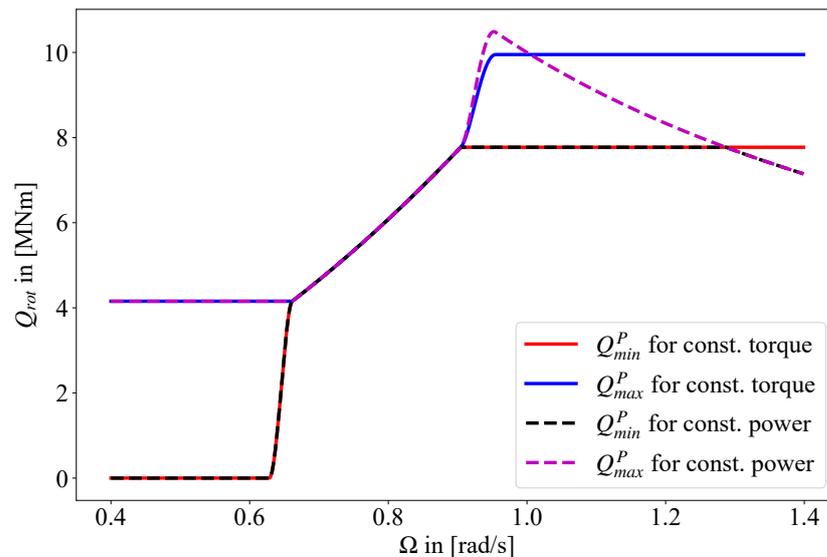


Figure 3. Limits for torque controller at the partial load regime. Limits are shown for the constant power and constant torque strategies.

Below the minimum rotor speed, Q_{min}^P is kept at 0 kNm to allow the rotor to gain enough rotational speed at low wind speeds. Once the rotor speed reaches Ω_{min-a} (95% of Ω_{min} in the figure), Q_{min}^P is increased and equaled to $K_{opt} \cdot \Omega^2$. This enforces the optimal torque-speed curve of the wind turbine below the rated speed. Once Ω gets close to Ω_{max-a} (95% of Ω_{rated}), Q_{max}^P opens up to allow the PID controller to regulate the rotor speed. The parameters Ω_{min-a} , Ω_{min-b} , Ω_{max-a} and Ω_{max-b} are externally defined by the user. Figure 3 includes the behavior of the torque limits for the two torque control strategies in the full load regime: constant power and constant torque. These are explained in the next section.

3.1.2. Full Load Regime

The full load regime of the controller uses the same PID controller for the controller torque, but now the torque limits Q_{min}^P and Q_{max}^P take the same value Q_{ref}^F . The control

strategy of the torque controller in the full load regime can be either constant power or constant torque. Q_{ref}^F takes different values depending on the strategy. These are:

$$Q_{ref}^F = \begin{cases} \frac{P_{rated}}{\Omega_{rated}} & \text{for constant torque} \\ \frac{P_{rated}}{\Omega} & \text{for constant power.} \end{cases} \quad (8)$$

While Q_{ref}^F is constant for the constant torque strategy, it is inversely proportional to the (unfiltered) rotor speed signal for the constant power strategy. This explains the $1/\Omega$ behavior of Q_{min}^P and Q_{max}^P for the constant power strategy in Figure 3.

3.1.3. Switching between Regimes

The controller switches between partial and full load regimes depending on the first order low-pass filtered switch variable $\sigma_{\theta-LP}$. σ_{θ} is defined using Equation (A5) as $\sigma(\theta_{f1}, \theta_{f2}, \theta_{mean})$. The limits θ_{f1} and θ_{f2} are user-defined parameters. For this study, the limits were defined as $\theta_{f1} = \theta_{f2} = \theta_{min} + 0.5^\circ$. In this configuration, the switching function σ_{θ} behaves like a step function. The variable θ_{min} is the minimum pitch angle of the pitch controller, defined in Section 3.2. The equation of the first order low-passed filter is given by (A1).

Including the switching between regimes and the torque strategy in the above rated regime, the full torque limits take the form

$$Q_{min} = (1 - \sigma_{\theta-LP}) \cdot Q_{min}^P + \sigma_{\theta-LP} \cdot Q_{ref}^F \quad (9)$$

$$Q_{max} = (1 - \sigma_{\theta-LP}) \cdot Q_{max}^P + \sigma_{\theta-LP} \cdot Q_{ref}^F. \quad (10)$$

3.1.4. Drivetrain Damper

The torque controller also includes a drivetrain damper to damp unwanted oscillation of the turbine shaft due to e.g., excitation of its torsional eigenfrequency. The drivetrain damper is implemented as a proportional term k_{dmp} to the band-passed rotor speed signal Ω_{BPF} :

$$Q_{dmp} = k_{dmp} \cdot \Omega_{BPF}. \quad (11)$$

The equation of the band-pass filter is given in Equation (A3).

Q_{dmp} is added to Q_{PID} to obtain the total output torque of the controller:

$$Q_{gen}(t) = Q_{dmp}(t) + Q_{PID}(t). \quad (12)$$

3.1.5. Disable for Low Rotor Speeds

An additional feature of the controller is a switch-off mechanism for low rotor speeds. The rotor speed signal Ω is filtered using a low pass filter (see Equation (A2)) and compared to a fraction of the minimum rotor speed: $k_{off} \cdot \Omega_{Min}$. If the filtered rotor speed signal is below this value, then the torque controller shuts down by setting $Q_{max} = Q_{min} = 0$ Nm. Note that if the turbine rotor speeds up again and the (filtered) rotor signal exceeds the aforementioned limit, then the torque controller is activated again and functions as described above.

3.2. Pitch Controller

The pitch controller in TUBCon features two strategies. The first is the Collective Pitch Control (CPC) strategy for power regulation. The second is the advanced Individual Pitch Control (IPC) strategy for additional load reduction. The CPC strategy is always active while the IPC strategy can be enabled by the user through the external parameter file.

3.2.1. Collective Pitch Control

The CPC controller combines two PID controllers that react to the rotor speed error and the power error. The inputs for the pitch controller are the rotor speed Ω , the mean

pitch angle θ_{mean} , the reference power from the torque controller P_{ref} and the measured wind speed at hub height V_{hub} (see Figure 2(a)).

The rotor speed signal is filtered using a second order low-pass filter as described by Equation (A2). Both the pitch and the torque controller share the same filter frequency and damping ratio to calculate Ω_{LP} . The speed error e_{Ω} is calculated as the difference between the low pass filtered rotor speed Ω_{LP} and the rated rotor speed Ω_{rated} . The error e_{Ω} is then passed through a notch filter to filter out the drivetrain eigenfrequency. The equation of a notch filter is given in Equation (A4).

In parallel, the power error e_P is calculated as the difference between P_{ref} and P_{rated} and passed through a notch filter (Equation (A4)) to obtain e_{P-NF} . The proportional, integral and differential terms of the pitch controller are calculated as follows:

$$\theta_P(t) = \frac{1}{2}(k_{P-\Omega} \cdot e_{\Omega-NF}(t) + k_{P-P} \cdot e_{P-NF}(t)), \quad (13)$$

$$\theta_I(t) = \frac{1}{2} \left(\int_0^t (k_{I-\Omega} \cdot e_{\Omega-NF}(\tau) + k_{I-P} \cdot e_{P-NF}(\tau)) d\tau \right), \quad (14)$$

$$\theta_D(t) = k_{D-\Omega} \cdot \frac{de_{\Omega-NF}(t)}{dt}. \quad (15)$$

Here, k_{P-X} , k_{I-X} and k_{D-X} are the proportional, integral and derivative constants of each PID controller. The subscript Ω denotes that the PID constants are applied to $e_{\Omega-NF}$. Likewise, the subscript P denotes the affiliation of the constants to e_{P-NF} . Note that k_{D-P} is always set to zero.

Additionally, the pitch signal is gain-scheduled with two factors. The first one accounts for the non-linear effect that larger blade pitch angles have on the aerodynamic torque. The second factor increases sensitivity of the pitch controller to large speed excursions in order to limit them. The total gain schedule is calculated via

$$\eta = \eta_{\theta} \cdot \eta_{\Omega} = \left(\frac{1}{1 + \frac{\theta_{LP}}{K_1} + \frac{\theta_{LP}^2}{K_2}} \right) \cdot \left(1 + \frac{e_{\Omega}^2}{(\Omega_2 - \Omega_{rated})^2} \right). \quad (16)$$

In this equation, θ_{LP} is the low-pass filtered mean pitch signal θ_{mean} . K_1 , K_2 and Ω_2 are parameters given by the user. To calculate θ_{LP} , a first order filter as described in Equation (A1) is used.

The pitch angle signal is limited by the parameters θ_{max} and θ_{min} . In addition, θ_{min} can be modified by V_{Hub} using a look-up table given by the user. In this case, the measured wind speed at hub height is also low-passed using a first order low pass filter (Equation (A1)).

In order for the pitch controller to react quickly if the required pitch signal changes suddenly from a saturated value, an anti-windup scheme like the one described in Equation (3) is used for θ_I .

The total collective pitch angle signal of the pitch controller is therefore given by

$$\theta_{PC} = \begin{cases} \eta \cdot (\theta_P + \theta_I + \theta_D) & \text{if } \theta_{min} \leq \theta_{PC} \leq \theta_{max} \\ \theta_{min}(V_{hub}) & \text{if } \theta_{PC} < \theta_{min}(V_{hub}) \\ \theta_{max} & \text{if } \theta_{PC} > \theta_{max}. \end{cases} \quad (17)$$

Additionally, the pitch rate is also limited by two user defined parameters: $\dot{\theta}_{max}$ and $\dot{\theta}_{min}$.

The power dependency of θ_{PC} will make the pitch controller raise the pitch angle set point when P_{ref} from the torque controller is close to P_{rated} . This in turn helps trigger the switching procedure from partial to full load regime of the torque controller as explained in Section 3.1.3.

3.2.2. Individual Pitch Control

One of the most common advanced pitch control strategies in wind energy research is the Individual Pitch Control (IPC) strategy [3,33]. Several studies have used this strategy as a comparison to other advanced pitch controller strategies [4,11,34]. It has also been used as a comparison or a complementary strategy for trailing edge flap controllers [7,9,10].

Figure 4 shows the graphical representation of the IPC strategy implemented in TUBCon.

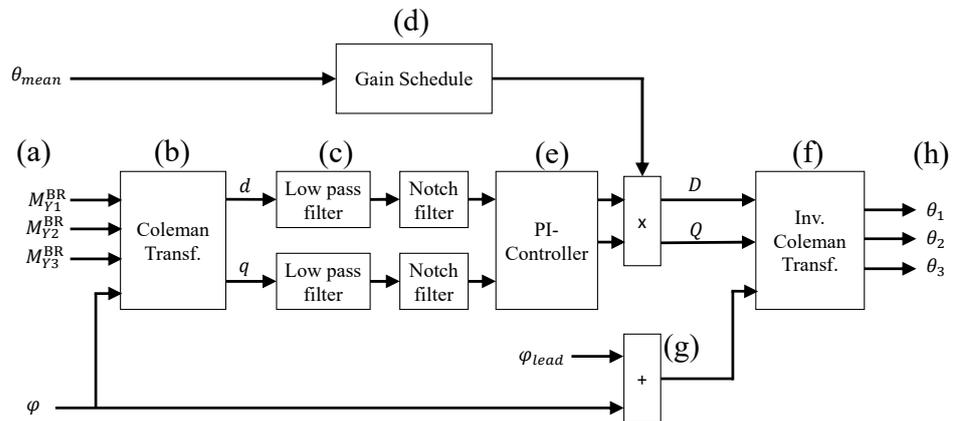


Figure 4. Graphical representation of the Individual Pitch Control (IPC) strategy implemented in the controller. The Blade Root Bending Moments (BRBM) signals are transformed into a non-rotating coordinate system and filtered using a low-pass and a notch filter. The filtered signals are then used in a PI controller to calculate the control signals. These signals are transformed back into the rotating frame of reference to be used as input for the individual pitch angles.

The IPC uses as input signals the out-of-plane BRBM of the three blades (M_{Y1}^{BR} , M_{Y2}^{BR} , M_{Y3}^{BR}) as well as the mean pitch angle θ_{mean} and the rotor azimuth angle φ (Figure 4(a)). The out-of-plane BRBMs are transformed to the direct and quadrature axes using the once-per-revolution or 1P-Coleman transformation (Figure 4(b)). The 1P-Coleman transformation for a three bladed rotor takes the form

$$\begin{pmatrix} d \\ q \end{pmatrix} = \frac{2}{3} \begin{pmatrix} \cos(\varphi) & \cos(\varphi + \frac{2\pi}{3}) & \cos(\varphi + \frac{4\pi}{3}) \\ \sin(\varphi) & \sin(\varphi + \frac{2\pi}{3}) & \sin(\varphi + \frac{4\pi}{3}) \end{pmatrix} \begin{pmatrix} M_{Y1}^{BR} \\ M_{Y2}^{BR} \\ M_{Y3}^{BR} \end{pmatrix}, \quad (18)$$

where d and q are the quantities expressed in the direct and quadrature axes respectively. The physical interpretation of these quantities is the rotor tilt and yaw moment in the non-rotating coordinate system.

These rotor moments are then passed through a second order low-pass filter (Equation (A2)) and a notch filter (Equation (A4)) Figure 4(c) to filter out unwanted high frequency content as well as the 1P component of the load signals in the non-rotating frame of reference. This step is needed as a 1P oscillation in the non-rotating frame of reference can lead to an increased 3P excitation of the turbine [35]. Following a concept presented in [9], a gain schedule is implemented using θ_{mean} to adapt the individual pitching action to the wind speed (Figure 4(d)). The gain schedule has the same mathematical expression as η_{θ} in Equation (16) but uses different parameter values. A PI controller is implemented with a zero set-point to reduce the rotor tilt and yaw moments (Figure 4(e)).

The control demand is transformed back to the demands of the individual pitch angles using the inverse 1P-Coleman transformation (Figure 4(f)). The inverse transformation is given by

$$\begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix} = \begin{pmatrix} \cos(\varphi') & \sin(\varphi') \\ \cos(\varphi' + \frac{2\pi}{3}) & \sin(\varphi' + \frac{2\pi}{3}) \\ \cos(\varphi' + \frac{4\pi}{3}) & \sin(\varphi' + \frac{4\pi}{3}) \end{pmatrix} \begin{pmatrix} D \\ Q \end{pmatrix}, \quad (19)$$

where the quantities D and Q represent the control signals in the non-rotating coordinate system and θ_1 , θ_2 and θ_3 the individual pitch angles for each blade (Figure 4(h)). The inverse Coleman transformation uses a modified azimuth angle φ' . Figure 4(g) shows that $\varphi' = \varphi + \varphi_{lead}$, where φ_{lead} is the lead angle. This constant angle helps decoupling the rotor yaw and tilt moments in the non-rotating frame of reference so that they can be treated as independent systems. As explained in [4], certain factors such as blade stiffness, collective pitch angle and pitch actuator response time introduce a dependency between the two rotor moments. Using the lead angle is a straightforward way to counter this problem.

It is possible to generalize this strategy to other frequencies by using nP Coleman transformations. In this case the rotor angle φ and the respective shift for each blade in Equations (18) and (19) are replaced by n -times their value. Using for example an IPC-like strategy with a 2P Coleman transform helps to reduce the 3P asymmetrical loads in the yaw bearing of the turbine [10].

The IPC pitch angles are limited to a maximum and minimum value ($\theta_{max-IPC}$ and $\theta_{min-IPC}$) given by the user. Since the PI controller determines the quantities D and Q in the non-rotating coordinate system, the limits are transformed into this coordinate system:

$$\frac{\theta_{min-IPC}}{\sqrt{2}} \leq D, Q \leq \frac{\theta_{max-IPC}}{\sqrt{2}}. \quad (20)$$

As with the CPC strategy, an anti-windup procedure equivalent to Equation (3) is used to limit the increase of the integral part of the controller.

The pitch angle signal of the IPC strategy is added to the collective pitch signal to obtain the final pitch angle signal. The use of this strategy increases the pitch activity significantly. In order to limit this increase in pitch activity, the control strategy is phased out in the partial load regime. Bergami and Gaunaa show in [36] that the majority of fatigue loading that can be alleviated with this strategy occurs in the above-rated region. Phasing out the IPC strategy helps limit the pitch activity and optimizes energy capture by keeping the pitch angle constant in the below-rated region. The phasing out is done by multiplying the IPC PI-constants with a switch function (Equation (A5)) based on θ_{mean} . The switch limits used in this study are $x_0 = 0.5^\circ$ and $x_1 = 3^\circ$.

3.3. Supervisory Control

The supervisory control of TUBCon has two main functions. The first one is to detect whenever some security limit (e.g., generator over-speed) has been tripped and enforce an appropriate maneuver (e.g., normal stop). The second function is to trigger specific events. These can be controller fault events (as required by current industry standards) or stop maneuvers (if a specific alarm has been triggered).

The control signals from the supervisory control will always override the signals from the pitch and torque controllers. This is shown in Figure 2 with the switch for each controller. In this figure, $\vec{\theta}_{SC}$ represents the pitch signal from the supervisory control that overrides the signal from the pitch controller. In the same manner, Q_{gen-SC} overrides the generator torque from the torque controller. In addition, the supervisory controller can activate the brake and the yaw actuator of the turbine.

The internal logic of the supervisory control is based on [37] and shown here in Figure 5. It features three control blocks. The first one checks if any turbine sensor is outside its allowable range and triggers an alarm should this happen (Figure 5(a)). The turbine sensors are bundled into the vector \vec{S}_{SC} in Figure 5 for the sake of clarity (see Section 3.3.1 for a full description of the considered sensors). Depending on the situation, the alarm can have several levels. An example of this could be the tripping of the first over-speed limit and the tripping of the second over-speed limit, both alarms depending on the sensor Ω .

The second block manages the alarms that the first block triggered and sets an appropriate reaction maneuver (Figure 5(b)). Keeping the example above, if the alarm block detected a first over-speed trigger, the managing block would request a normal stop of the turbine. If a second over-speed trigger is detected, then the manager block would request

an emergency stop procedure. The latter clearly has a higher priority than the former. Both stop procedures have a higher priority than the normal control action of the controller.

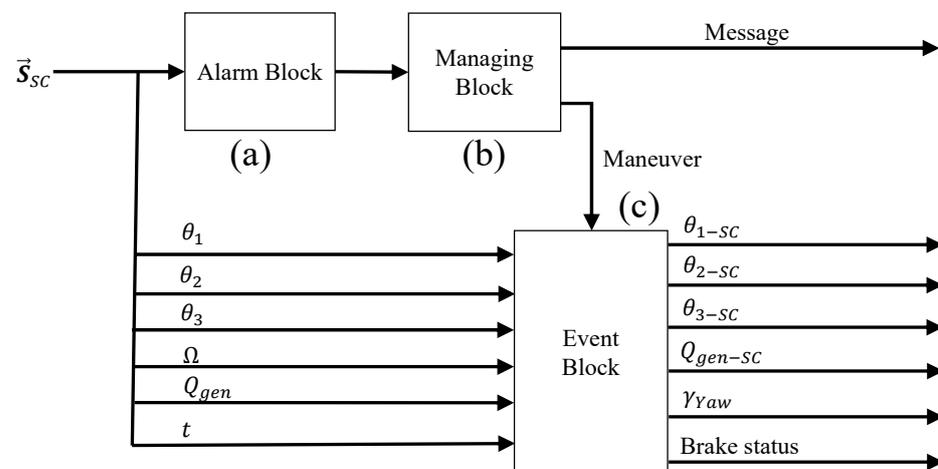


Figure 5. Overview of the supervisory control. It comprises an alarm block, a managing block and an event block. The latter can be used to trigger specific events such as controller faults that are required for a full load calculation.

The third block is the event generator block (Figure 5(c)). This block creates the necessary controller signals to simulate any given event. These signals could be a controller fault required by a specific design load case or the normal stop maneuver requested from the manager block. The input signals for this block are the current pitch angle of each blade θ_i , the rotor speed Ω and torque Q_{gen} and the current time t . The last sensor is needed if an event is to occur at a specific simulation time. The output of the event block are the overridden controller signals. Note that depending on the specific event, not all control signals are necessarily overridden.

3.3.1. Considered Alarms and Reaction Maneuvers

Table 1 lists all the considered alarms of the supervisory control. The first column shows the measured sensor, the second column the limit and the third column the requested maneuver. In the third column, NS stands for normal stop and ES for emergency stop. These are the only two currently implemented maneuvers of the controller and are explained below.

The alarm block considers five sensors: rotor speed Ω , electrical power P , yaw misalignment γ , pitch angles θ and grid status. Both Ω and P have two limits ($n4$, $p4$ and nA , pA respectively) that, when surpassed, trigger instantly the appropriate stop maneuver. The same simple trip-logic is implemented for the grid status signal. At the instant that the grid status is measured as offline, a normal stop maneuver is triggered.

For the other sensors, the alarm is triggered if the sensor surpasses the limit for a given time period. The alarm block foresees two limits for γ with two different duration times. The first limit γ_{max-1} considers long-time yaw misalignments. The measured yaw misalignment γ has to be larger than this limit for a time interval greater than $\tau_{\gamma-1}$, which should be fairly large. In contrast, γ_{max-2} considers short-term extreme yaw misalignments. The yaw misalignment signal used to trigger the alarms is filtered with a first order low pass filter (Equation (A1)) to even out variations coming from inflow turbulence.

Table 1. Limits and reaction maneuvers of the supervisory control. Depending on the pitch control strategy, only $\Delta\theta_{ij}$ or $\Delta\theta_{i-CP}$ can be active. NS = normal stop; ES = emergency stop.

Sensor	Limit	Maneuver
Ω	$>n4$	NS
Ω	$>nA$	ES
P	$>p4$	NS
P	$>pA$	ES
γ	$>\gamma_{max-1}$ for $\tau_{\gamma-1}$ seconds	NS
γ	$>\gamma_{max-2}$ for $\tau_{\gamma-2}$ seconds	NS
$\Delta\theta_{ij}$	$>\Delta\theta_{max-CPC}$ for $\tau_{\Delta\theta-CPC}$ seconds	NS
$\Delta\theta_{i-CP}$	$>\Delta\theta_{max-IPC}$ for $\tau_{\Delta\theta-IPC}$ seconds	NS
$\Delta\theta_{set}$	$>\Delta\theta_{set-max}$ for $\tau_{\Delta\theta-set}$ seconds	NS
Grid	Offline	NS

For the pitch angle group, there are two conditions that can trigger alarms. The first one is the difference between the individual pitch angles ($\Delta\theta_{ij}$ or $\Delta\theta_{i-CP}$). This difference is calculated differently depending on the active pitch control strategy. For the CPC strategy (Section 3.2.1), the alarm block checks the difference between all three angles. If the difference between any two pitch angles is greater than the fixed limit $\Delta\theta_{max-CPC}$ for a given time $\tau_{\Delta\theta-CPC}$, then the alarm is triggered and a normal stop maneuver is requested. For the IPC strategy (Section 3.2.2), the pitch angle differences are measured against the collective pitch angle signal. The limit $\Delta\theta_{max-IPC}$ is not constant but varies depending on the short-term mean difference between the IPC angle and the collective pitch angle (plus a user-defined offset angle). The short-term mean difference of the individual pitch angles is obtained by applying a first-order low pass filter to the instantaneous mean difference. The time constant for this filter is a user-defined parameter. By using this triggering strategy, the supervisory control can detect pitch angle faults even when the difference between two pitch angles lies below the maximum allowed pitch difference for the IPC strategy.

The second condition included in the pitch angle alarm group is the difference between the measured and the demanded pitch angle for each blade $\Delta\theta_{set}$. If the difference between the two signals in any blade is greater than $\Delta\theta_{set-max}$ for a given time $\tau_{\Delta\theta-set}$, then the alarm is triggered and a stop maneuver is requested.

The two stop maneuvers used in this controller are defined by a sequence of actions that are taken (almost) independently of the controller signals. The only exception is the normal stop maneuver that additionally uses the pitch angle differences, as explained below. The parameters for both maneuvers are listed and explained in Table 2. Note that there are two sets of parameters: one for the normal stop and one for the emergency stop procedure.

The implemented stop maneuvers foresee that the collective pitch angle will increase at a rate $\dot{\theta}_S$ up to a maximum value θ_{max-S} . This pitching towards feather can be delayed for a given time using the parameter τ_θ . Parallel to the pitch controller, the torque controller decreases linearly to Q_{min-S} at a rate given by the parameter \dot{Q}_S . This procedure can also be delayed for a given time using the parameter τ_Q . In case of a grid loss, a chopper can keep the generator torque constant for a given time $\tau_{Chopper}$ in order to avoid very high rotor speeds. There is also the option to enable the turbine brake during the stop maneuvers. This is controlled with an enable flag and a minimum rotor speed ($\Omega_{Brake-S}$) at which the brake is activated.

In addition to these fixed parameters, the normal stop procedure checks the pitch angle differences between all blades. If the absolute value of a given difference is above 0.5° , then the pitch rate $\dot{\theta}_S$ is replaced by the minimum value between $\dot{\theta}_{max}$ and $1.5 \cdot \dot{\theta}_S$. This is done for all blades except the one with the highest pitch angle. This way the pitch angle differences are minimized during the stop maneuver avoiding unnecessary oscillations due to aerodynamic imbalances of the rotor. The emergency stop does not have this feature as it is designed to be hard coded in the turbine's safety mechanism.

Table 2. Parameters for the normal and emergency stop maneuvers.

Parameter	Explanation
\dot{Q}_S	Torque rate during stop.
Q_{max-S}	Maximum allowed torque during stop.
Q_{min-S}	Minimum torque during stop. Generator torque ramps to this value.
τ_Q	Delay time at which the torque is kept constant before ramping down.
$\tau_{Chopper}$	Delay time at which the torque is kept constant in case of grid loss (only NS).
$\dot{\theta}_S$	Collective pitch rate during stop.
θ_{max-S}	Maximum allowed pitch angle during stop.
θ_{min-S}	Minimum allowed pitch angle during stop.
τ_θ	Delay time at which the pitch angle is kept constant before pitching.
BrakeFlag _S	Flag to enable the brake in the maneuver
$\Omega_{Brake-S}$	Rotor speed at which the brake is enabled

3.3.2. Considered Events

The event block allows the controller to simulate the events required by the current industry standards for a full load calculation [21]. Table 3 presents and explains all the events that the controller is able to perform. All of the listed events are controlled by a set of parameters passed to the controller.

Table 3. Events included in the event block.

Name	Explanation
Normal stop	Trigger normal stop maneuver.
Emergency stop	Trigger emergency stop maneuver.
Pitch runaway to feather/fine	Runaway of one or multiple blades to the 90°/0° position.
Speed transducer fault	Generator speed signal is erroneously measured.
Grid loss	Sudden loss of the electrical grid.
Yaw runaway	Runaway of the yaw motor.

Both the normal stop and the emergency stop events require two parameters: a flag to indicate if the event will be triggered and the time at which the maneuver is triggered.

The pitch runaway to feather/fine fault simulates an error in one or several pitch actuators. In this event, one or several blades suddenly pitch with a given pitch rate to the respective extreme value. The pitch rate is one of the parameters for this event. The sign of the pitch rate defines if the event is a pitch-to-feather (positive rate) or a pitch-to-fine (negative rate) fault event. There is one activation flag and one activation time parameter for each blade. This gives the flexibility to simulate the single pitch fault events and the collective pitch fault events with a limited set of parameters.

The pitch transducer fault simulates an error in the measurement of the generator speed. Instead of the actual speed, the transducer measures a constant value and passes it to the pitch and torque controller. The parameters for this event are a flag to enable the event, the time at which the event starts and the constant generator speed passed to the controller once the fault occurs.

The grid loss event simulates a sudden loss of the electrical grid. As a consequence, the generator cannot feed the generated power to the grid and the torque drops to 0 Nm. This event also raises the grid loss flag so that the alarm block recognizes the event and triggers the appropriate maneuver. The parameters that control this event are a flag that enables it and the time at which the grid loss event happens.

Finally, the yaw runaway event simulates a malfunction in the yaw system that causes the yaw motor to activate. The parameters for this event are an activation flag, a time to start yawing, a yaw rate and a time to stop yawing.

3.4. Actuator Models

Once the set-point values of the blade pitch angles and generator torque are calculated by the respective controllers, they are passed to the actuator models of the controller (see Figure 2). The actuator models are used to model the additional inertial effects of the physical actuators. These models can be very accurate depending on the individual needs of the calculation.

For the TUB Controller, a simple second-order linear model was chosen to model both the pitch actuators and the generator model. The second-order model has the same mathematical description as a second-order low-pass filter, namely Equation (A2). This model is also used for the yaw actuator in the yaw runaway event. The individual cut-off frequencies and damping ratios for the actuators and the generator model are supplied as parameters by the user. For the brake actuator, a simple first-order linear model as described by Equation (A1) is used. The time constant is supplied by the user.

3.5. Supervisory Control Event Examples

Figure 6 shows four examples of events and stop maneuvers using the DTU 10 MW RWT in turbulent wind simulations. In Figure 6a, a collective pitch-to-fine fault at a simulation time of 125 s leads to the triggering of the first and second over-speed alarms. The resulting maneuvers (normal and emergency stop) can be distinguished by the different pitch rates. Figure 6b shows a single pitch-to-fine event during power production using the IPC strategy. At 120 s of simulation time, θ_1 receives a faulty signal and starts pitching towards 0° . This is caught by the supervisory controller and the normal stop procedure is triggered. The normal stop feature that minimizes aerodynamic imbalances can be clearly seen in this figure, as the difference in individual pitch angles is reduced to 0° at 130 s of simulation time. Figure 6c shows a grid loss scenario and the resulting stop maneuver. Finally, Figure 6d shows a speed transducer fault at 120 s simulation time. As a consequence, the generator torque and pitch angle signals freeze and a normal stop is only triggered because of the first over-power alarm (comparable to the first over-speed threshold).

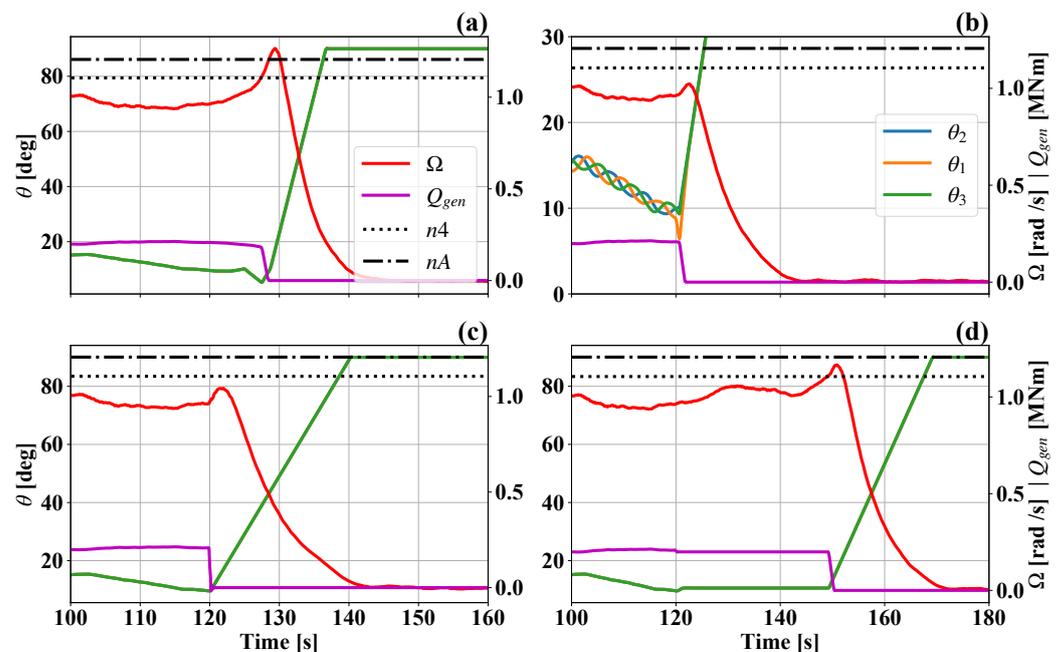


Figure 6. Events triggered by the controller: (a) Collective pitch-to-fine; (b) Individual pitch-to-fine; (c) Grid loss; (d) Speed transducer fault. In all cases, the event is captured by the supervisory controller and an appropriate stop maneuver is triggered.

4. Controller Validation

The TUB Controller was tested and validated in a series of load cases to verify its functionality. This was done in steady and turbulent wind conditions, summarized in Table 4. The validation was done by comparing TUBCon to the established Basic DTU Wind Energy Controller [16]. The parameters for both controllers were taken from the LIFE50+ report [38].

Table 4. Main Parameters of the Simulations used for Controller Verification.

Parameter	Steady Calculations	Turbulent Calculations
Mean V_{Hub}	4–24 m/s	4–24 m/s
Wind model	steady	IEC NTM
Wind shear exp.	0	0.2
Upflow angle	0°	8°
Yaw angle	0°	−8°, 0°, 8°
Sim. time	350 s	600 s

4.1. Steady Wind Simulations

Figure 7 shows the behavior of both controllers in the steady wind simulation.

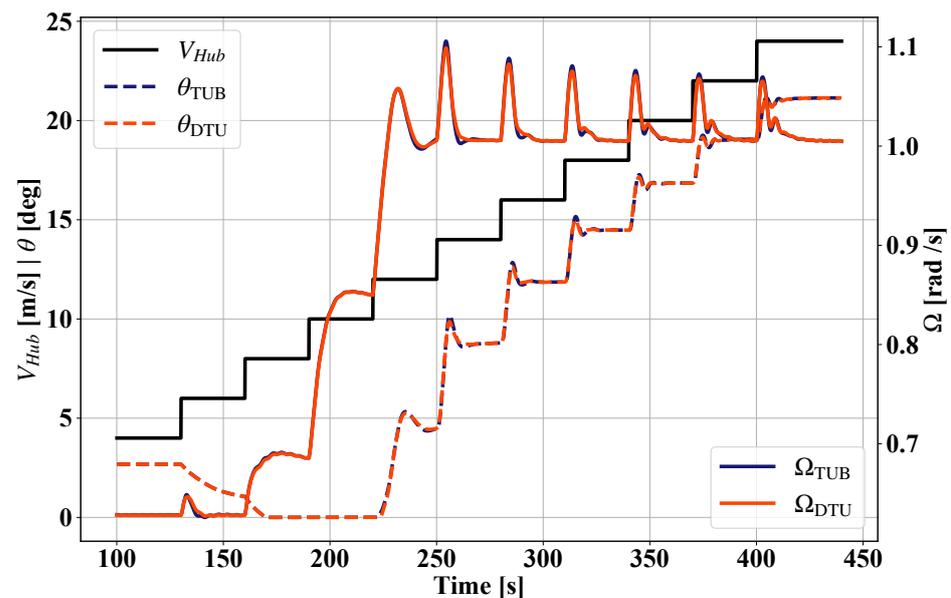


Figure 7. Validation of the controller in steady wind simulations. Pitch angle θ and rotor speed Ω of both controllers are compared for ascending wind steps.

This simulation features a series of wind speed steps from 4 m/s to 24 m/s with a 2 m/s step height. Between two steps, the wind speed is held constant for 30 s. It can be seen that both controllers behave in an almost identical manner. This is due to the fact that the collective pitch and torque controllers from TUBCon were directly adapted from the Basic DTU Controller. With the chosen parameters, TUBCon manages to stabilize the turbine without large overshoots in the pitch and rotor speed signals in spite of the sudden increases in wind speed. This is true for all simulated wind speeds.

4.2. Turbulent Wind Simulations

The turbulent wind simulations were done using the Normal Turbulent wind Model (NTM) from [21]. Because of this, the controller validation has to be done in a statistical manner. The setup is given in Table 4 under the column “Turbulent calculations”. For each wind speed bin, we did six simulations with three different yaw angles of the turbine and

two turbulence seeds. In total, there were 66 simulations, each 600 s in length for each controller. Because we are using QBlade's LLFVW method, the wake needs to first develop behind the rotor in order to have accurate results. We added 200 s of pre-simulation time to allow the wake to develop. This time was discarded in the analysis. Figure 8 shows an example of an aero-servo-elastic simulation of the DTU 10 MW RWT within QBlade. The incoming turbulent wind speed has an average hub-height value of 14 m/s and is shown on the left. The wake is modeled by vortex elements that are allowed to convect freely downstream of the turbine.

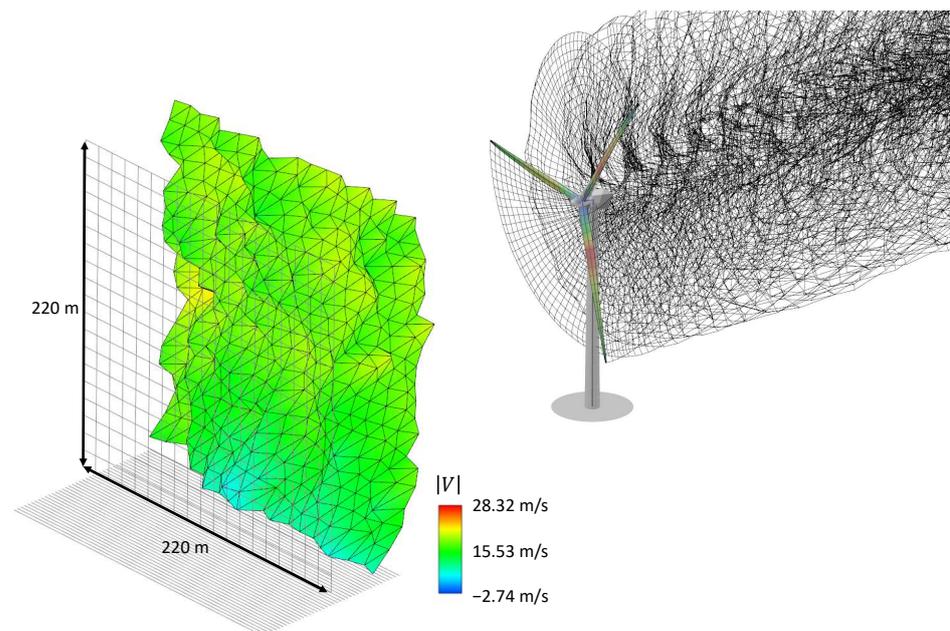


Figure 8. Turbulent wind aero-servo-elastic simulation with QBlade's Lifting Line Free Vortex Wake (LLFVW) aerodynamic model. The DTU 10 MW Reference Wind Turbine (RWT) is simulated in a turbulent wind field with an average hub-height wind speed of 14 m/s.

To compare the controllers we chose a selection of turbine sensors that are summarized in Table 5. These sensors give a good estimate of the overall loading of the turbine blades, tower top and tower bottom.

Table 5. Considered sensors for turbulent wind simulations

Sensor Name	Symbol
Blade pitch angle/rotor speed/generator power	$\theta/\Omega/P$
Blade root in-plane/out-of-plane/torsional bending moment	$M_X^{BR}/M_Y^{BR}/M_Z^{BR}$
Yaw bearing roll/tilt/yaw moment	$M_X^{YB}/M_Y^{YB}/M_Z^{YB}$
Tower base side-side/fore-aft/torsional bending moment	$M_X^{TB}/M_Y^{TB}/M_Z^{TB}$

We used two metrics to analyze the performance of the controllers, depending on the nature of the sensor. For the load sensors our metric is the lifetime Damage Equivalent Loads (DELs). These are calculated using the rainflow counting algorithm combined with the Palmgren-Miner linear damage accumulation hypothesis, as described in [39]. The rainflow count was done using the software Crunch [40]. For the controller signals, we use the averaged standard deviation of each signal. The average is taken from the individual standard deviations of all simulations in each of the wind speed bins. To calculate the lifetime DELs we used a wind speed distribution that corresponds to a wind class IA turbine [21], which is the design wind class of the DTU 10MW RWT.

4.2.1. Controller Signals

Figure 9 shows the averaged normalized standard deviation of the controller signals as a function of the average wind speed for the calculations with the DTU and the TUB controllers. The normalization is done with respect to the calculations with the DTU controller. We can see in this figure that the averaged standard deviations of the rotor speed and pitch angle are very similar with both controllers. The main difference is seen for Ω in wind speed bins of 12 m/s and higher. Here, the $\bar{\sigma}(\Omega)$ of the calculations with TUBCon are consistently higher than with the DTU controller. The reason for this can be seen in Figure 9c. Here we can see that $\bar{\sigma}(P)$ is considerably smaller for the TUBCon calculations compared to the calculations with the DTU controller. The torque controller objective was constant power for both controllers. The fact that $\bar{\sigma}(P)$ is smaller in the TUBCon simulations indicates a more aggressive constant power enforcement of this controller. This comes at the expense of larger torque and rotor speed fluctuations when compared to the calculations with the DTU controller. We note that even though the differences in the normalized $\bar{\sigma}(P)$ are large, the absolute differences in the variation of the generator power were small. The large relative differences seen in wind speed bins of 14 m/s and higher arise because the generator power is mostly constant and the values of $\bar{\sigma}(P)$ are small for both controllers.

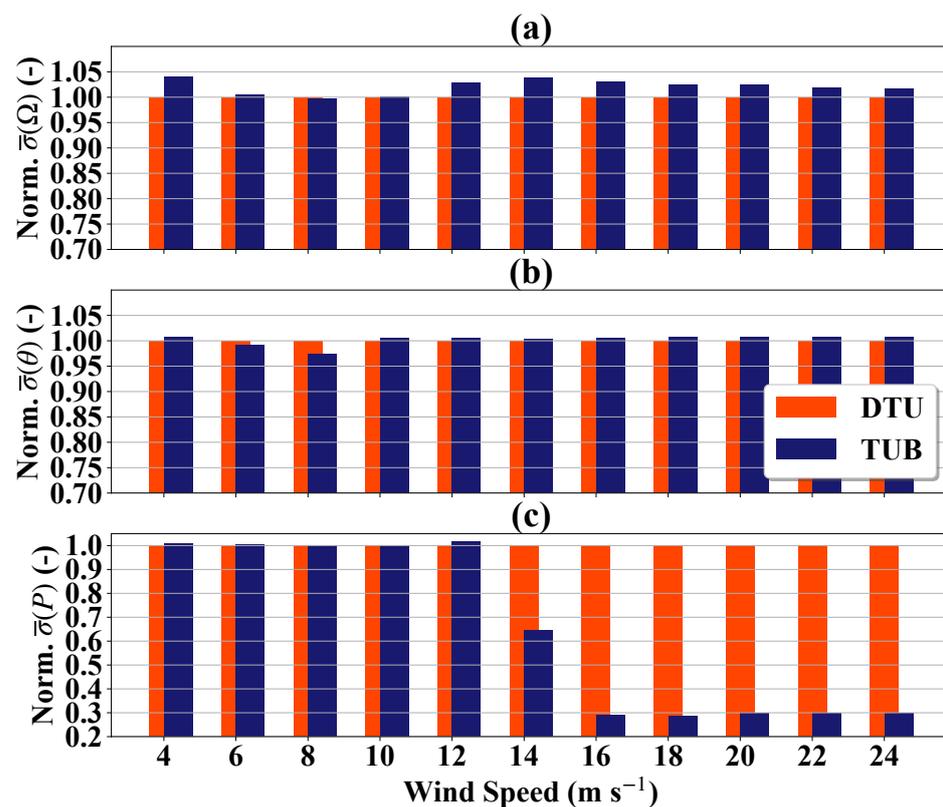


Figure 9. Normalized average standard deviations of controller signals vs. wind speed. (a) Rotor speed; (b) Pitch angle; (c) Generator power. Nomenclature of the signals is found in Table 5.

4.2.2. Fatigue Loads

The effect on the fatigue loads can be seen in Figure 10. The only significant differences occur for M_X^{TB} and M_Y^{TB} , with TUBCon simulations showing a 3% and 2% increase respectively, compared to simulations with the DTU controller. The differences in M_Y^{TB} come from the larger oscillations of Ω in the TUBCon simulations, driven by the more aggressive constant power controller. The increased fatigue loads of M_X^{TB} are only indirectly affected by controller action and it is therefore more difficult to find the source of this difference. It is assumed that the different variations of Ω lead to differences in the induced tower

oscillations. For the fore-aft direction, the tower vibrations are damped out quickly due to aerodynamic damping. This is not the case in the side-side direction as the damping is much lower and hence these vibrations cause different amounts of M_X^{TB} fatigue loads. Further investigation is needed to corroborate this assumption.

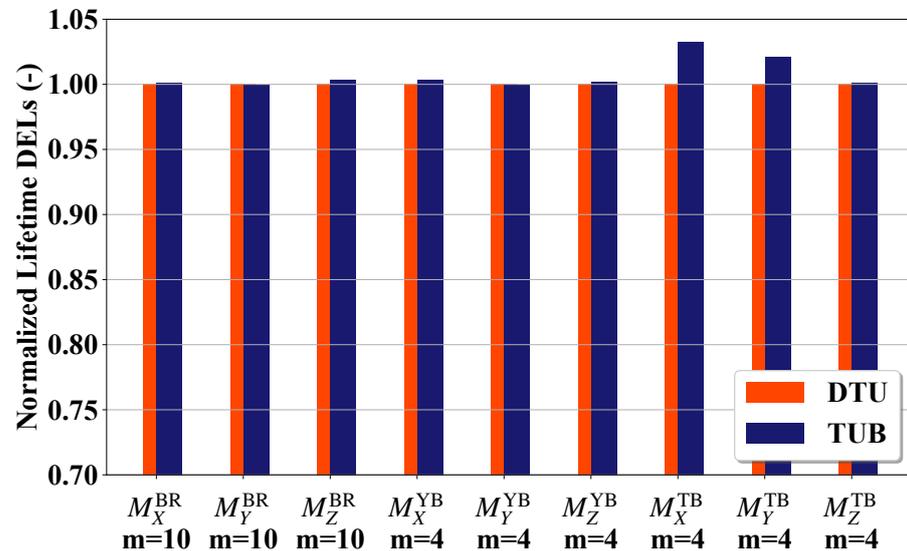


Figure 10. Normalized lifetime Damage Equivalent Loads (DELs) for considered sensors. Nomenclature is found in Table 5.

All in all the differences in controller behavior and turbine loading remain small and the performance of both controllers is very similar.

5. Fatigue Load Reduction through Advanced Control Action

As discussed in [27], many implementations of the BEM aerodynamic model average the axial induction factor across the turbine rotor. This leads to inaccuracies in the local induced velocities on the blades and hence to significant loading differences when compared to the more accurate LLFVW aerodynamic method.

These differences will necessarily affect the performance of advanced control strategies such as IPC. Firstly, the input signals of the IPC controller will be different in LLFVW simulations because of the more accurate estimation of M_Y^{BR} in each blade compared to BEM-based simulations. Secondly, the controller action will affect the turbine loads differently due to the more accurate representation of the non-homogeneous induction field in LLFVW simulations caused by the individual pitch positions.

In this section we explore the load reduction potential of the IPC control strategy (Section 3.2.2) using the LLFVW method from QBlade. This will give us a more accurate picture of the load reduction capabilities of this strategy. The IPC strategy focuses on reducing the once-per-revolution (1P) M_Y^{BR} loads. The parameters were taken from the report [9] and slightly adapted to account for the different coordinate system in which the input is measured. Since we are interested mainly in fatigue load reduction, we considered the same load cases from Table 4 (column “Turbulent calculations”). These load cases correspond to the DLC group 1.2 from [21]. For onshore wind turbines, the DLC group 1.2 is the main contributor of lifetime fatigue loads for most of the turbine components, since the turbine spends most of its operating time in these conditions [41]. Evaluating the fatigue loads from this group will therefore give a close estimate of the real fatigue loads. To keep consistency, we considered the same load sensors and metrics as the ones described in Section 4.2.

5.1. Results

Figure 11 shows a selection of representative time series from a baseline simulation and a simulation with active IPC strategy. Both simulations use the TUB Controller with the same controller parameters. The baseline simulation uses the CPC strategy and the IPC simulation uses the CPC and IPC strategies. In this figure we can see the effect of the IPC strategy on the controller signals and on the M_Y^{BR} loads. From Figures 11b,e it can be seen that the influence of the IPC strategy on Ω and P is small. This is mainly due to the frequency separation between the CPC and IPC strategies. Figure 11c shows the time series of θ_1 . We can clearly see the 1P oscillation of the IPC strategy on top of the CPC pitch signal. The load reducing effect of this strategy can be seen in Figure 11d at around 700 s of simulation time. The 1P variation of M_Y^{BR} are clearly reduced due to the additional pitch actuation. The below-rated power limitation of the IPC strategy can be seen in Figure 11e around 650 s of simulation time. When the mean pitch angle decreases to 0° , the amplitude of the IPC signal is also reduced 0° to maximize power capture.

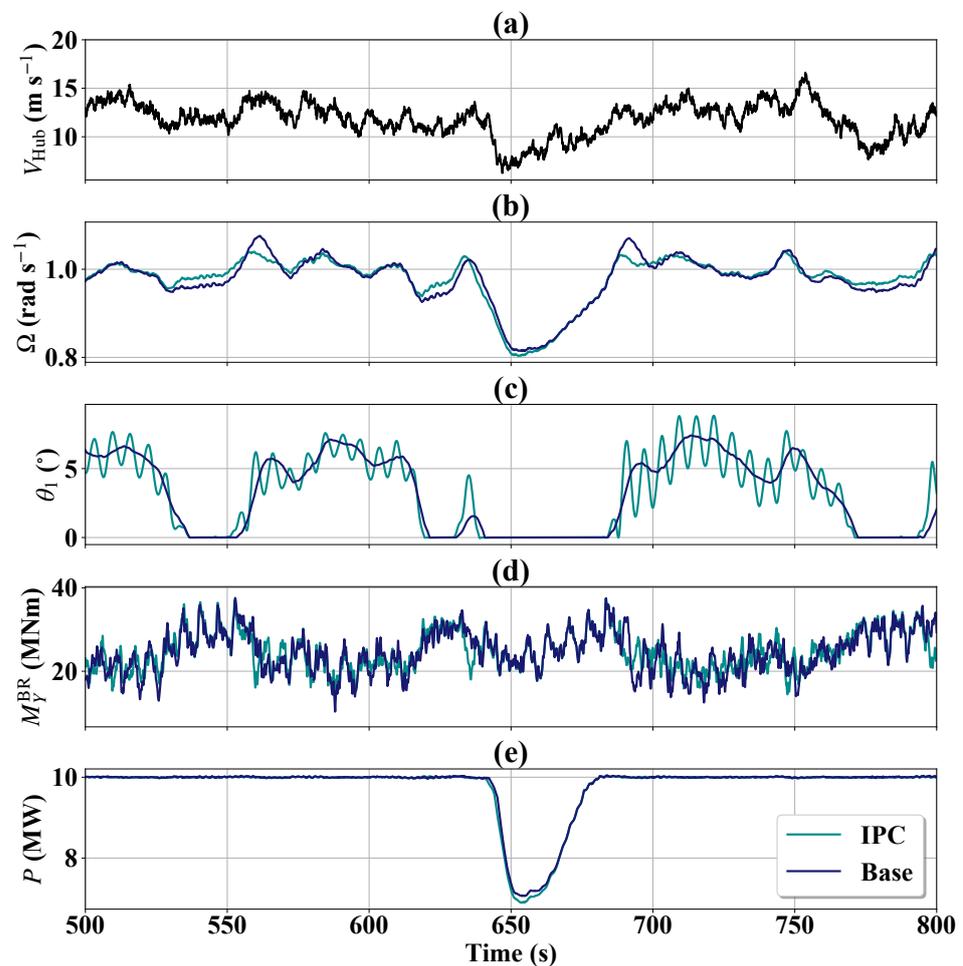


Figure 11. Time series selection of simulations with and without Individual Pitch Control (IPC) strategy from the 12 m/s wind speed bin. (a) Wind speed; (b) Rotor speed; (c) Blade 1 pitch angle; (d) Out-of-plane BRBM blade 1; (e) Generator power. Base = Collective Pitch Control (CPC) strategy.

5.1.1. Controller Signals

The effect of the IPC strategy on the controller signals can be seen in Figure 12. It shows the normalized averaged standard variations of Ω , θ and P respectively. Figure 12a shows that the normalized $\bar{\sigma}(\Omega)$ of the IPC strategy is practically 1 for all wind speed bins except the bins corresponding to 12 and 14 m/s average speed. There we see that Ω from the IPC simulations oscillates less than Ω from the CPC simulations. This can be understood

if we look at Figure 11b. The additional 1P fluctuation of θ does influence the value of Ω by reducing slightly the sensitivity of the rotor to changes in the wind speed. This is particularly marked around rated wind, where the pitch angle is close to 0° and there are high angles of attack in the outer span of the blades. The latter lead to large values of lift forces and changes in the lift force due to pitching have an increased effect on rotor thrust and torque.

Figure 12b shows the normalized $\bar{\sigma}(\theta)$ vs. the wind speed bin. This is the controller signal that shows the largest differences. From 10 m/s onward, the value of $\bar{\sigma}(\theta)$ increases up to values larger than 1.2. It is this region where the IPC strategy starts to function. The increase in pitch activity from the IPC strategy is directly linked to the fact that the 1P fluctuation of M_Y^{BR} increases with increasing wind speed [36].

Finally, Figure 12c shows the normalized values of $\bar{\sigma}(P)$. Here, there is almost no difference between the two strategies. The lower value in the IPC simulations seen in the wind speed bin of 14 m/s comes from the relatively small reference value of $\bar{\sigma}(P)$ for the CPC calculations. Because P is fairly constant in above-rated wind simulations, differences in the drop of P for temporarily low wind speeds (as seen in Figure 11e) will have a large effect on the normalized $\bar{\sigma}(P)$.

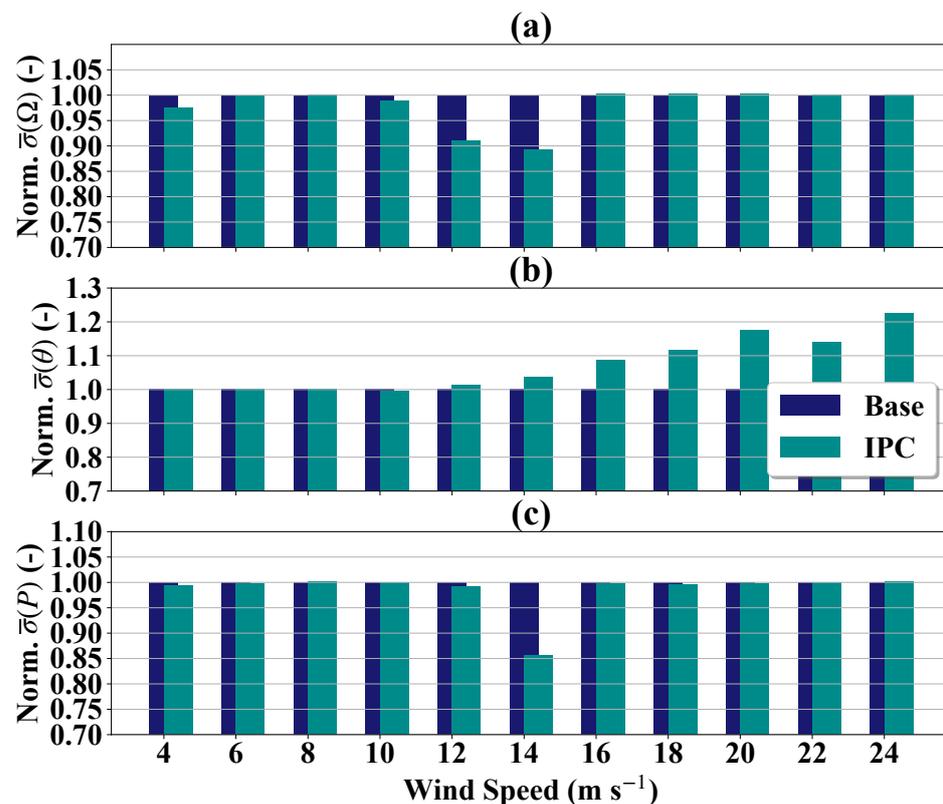


Figure 12. Averaged standard deviations of controller signals vs. wind speed. (a) Rotor speed; (b) Pitch angle; (c) Generator power. Nomenclature of the signals is found in Table 5. Base = CPC strategy.

5.1.2. Fatigue Loads

Figure 13 shows the normalized lifetime DELs of the simulations with the CPC and the IPC strategies. We can see that the main effect of the IPC strategy is to reduce the lifetime DELs of M_Y^{BR} and M_Z^{BR} by 14% and 9.4% respectively. There are also smaller effects on other load sensors. The lifetime DELs of M_X^{BR} are 2.4% lower in the simulations with the IPC strategy compared to the CPC strategy. The load differences for these three sensors come directly from the IPC strategy. Since the strategy is minimizing the yaw and tilt moments of the rotor, it indirectly minimizes the load fluctuations of the individual blade root moments.

These fluctuations arise from oscillations of the effective lift force on the individual blades. So, indirectly, the IPC strategy is also minimizing the fluctuation of the effective lift force on the individual blades. Although the out-of-plane component dominates in the lift force composition, there is also a smaller in-plane component. The former affects M_Y^{BR} and the latter M_X^{BR} . Reducing the lift fluctuations also reduces the pitching moment oscillations and oscillations in the out-of-plane deflection of the blades. These two quantities directly affect M_Z^{BR} . So by reducing the lift force fluctuation, all three load sensors are affected.

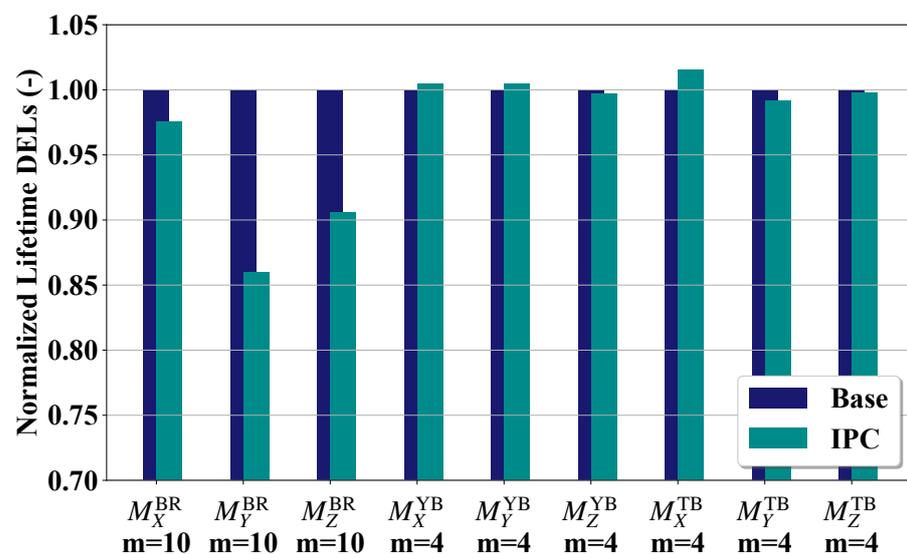


Figure 13. Normalized lifetime Damage Equivalent Loads (DELs) for considered sensors. Nomenclature is found in Table 5. Base = CPC strategy.

The yaw bearing sensors are practically unaffected by the IPC strategy. This is due to the nature of our metric. The 1P based IPC strategy focuses on reducing the steady or slow varying yaw and tilt moments of the rotor. Our calculation of the DELs does not consider the means but only the ranges of the load cycles. A reduction of the load averages is therefore not captured by our metric. An IPC strategy that additionally targets the 2P load fluctuations would also reduce the yaw bearing DELs, ([10,20], (pp. 501–503)). The tower base lifetime DELs show that the IPC strategy also has a small impact on the tower fatigue loads. The IPC strategy increases the M_X^{TB} DELs by 1.5% and decreases the M_Y^{TB} DELs by 0.9% compared to the CPC strategy.

6. Conclusions

This paper presented and detailed TUBCon, an advanced open-source wind turbine controller that includes pitch, torque and supervisory controllers. The controller can therefore be used to perform a complete load calculation according to industry standards. It is compatible with the common aeroelastic simulation codes including QBlade, which features the higher order LLFVW aerodynamic method in addition to a standard unsteady BEM aerodynamic model.

TUBCon was validated against an established turbine controller from the literature by comparing aeroelastic simulations of the DTU 10 MW RWT in steady and turbulent wind conditions. For steady wind conditions, the controllers show practically the same performance. For turbulent wind conditions, TUBCon shows a more aggressive constant power regulation. This reduces the mean standard deviation of the generator power for the above-rated wind bins by up to 70%, but increases the standard deviation of the rotor speed by up to 3.7%. This difference also affects the lifetime DELs of the tower base bending moments. The lifetime DELs of M_X^{TB} and M_Y^{TB} are increased in the TUBCon simulations by 3% and 2% respectively.

We also investigated the advanced load reduction capabilities of TUBCon by comparing the performances of the IPC strategy against a baseline CPC strategy. This was done using QBlade's LLFVW aerodynamic method, which is able to calculate the aerodynamic effects of a non-homogeneous induction field due to individual pitching more accurately compared to most BEM-based aerodynamic methods. The results show that the IPC strategy is able to reduce the lifetime DELs of M_Y^{BR} and M_Z^{BR} by 14% and 9.4% respectively when compared to the baseline simulations. This comes at the cost of increased pitch activity from the IPC controller. The normalized $\bar{\sigma}(\theta)$ of the IPC strategy increase to values up to 22.6% higher than the values of the CPC strategy for wind speed bins above rated wind speed.

Future work will include further development of TUBCon to add new features such as tower vibration damping and rotor speed exclusion. It is also planned to add a fully featured trailing edge flap controller and the necessary features so that TUBCon can be used as a controller in offshore floating wind turbine simulations. Furthermore, the controller will be extended to include power curtailment and wake steering capabilities so that it can be used in conjunction with a wind farm controller. In addition, the effect of QBlade's LLFVW aerodynamic model on advanced controller performance will be further analyzed by comparing the results from LLFVW-based simulations to more common BEM-based simulations and considering more DLC groups.

Author Contributions: Conceptualization, S.P.-B.; methodology, S.P.-B. and D.M.; software, S.P.-B. and D.M.; validation, S.P.-B. and D.M.; formal analysis, S.P.-B.; investigation, S.P.-B. and D.M.; resources, C.N.N. and C.O.P.; data curation, S.P.-B. and D.M.; writing—original draft preparation, S.P.-B.; writing—review and editing, D.M., C.N.N. and C.O.P.; visualization, S.P.-B.; supervision, C.N.N. and C.O.P.; project administration, C.N.N. and C.O.P.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: TUBCon and QBlade are open-source codes available online. The latest version of QBlade is available at <https://www.qblade.org>. The latest version of TUBCon is available at <https://github.com/s-perez-becker/TUBCon>. The time series for the load calculations are stored in the HAWC2 binary format. They can be made available upon request.

Acknowledgments: SPB wishes to thank WINDnovation Engineering Solutions GmbH for supporting his research.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Equations of Used Filters and Switches

This section includes the continuous functions of the used filters and switches. All filters are given in the Laplace domain. The equivalent form for discrete time can be found in [16].

The first order low-pass filtered value of a general variable is given by:

$$x_{LP}(s) = \frac{1}{\tau \cdot s + 1} \cdot x(s), \quad (A1)$$

where τ is the filter time constant.

The second order low-pass filter of a general signal takes the form

$$x_{LP}(s) = \frac{\omega^2}{s^2 + 2\zeta\omega \cdot s + \omega^2} \cdot x(s). \quad (A2)$$

Here x and x_{LP} denote the original and low-passed signal respectively. ω is the filter frequency and ζ the damping factor.

A band pass filter of a general variable is implemented as:

$$x_{BPF}(s) = \frac{2\zeta\omega \cdot (s + \tau \cdot s^2)}{s^2 + 2\zeta\omega \cdot s + \omega^2} \cdot x(s), \quad (A3)$$

where ω is the center frequency, ζ the damping ratio and τ a time constant. The notch filter of a general signal takes the form

$$x_{NF}(s) = \frac{s^2 + 2\zeta_2\omega \cdot s + \omega^2}{s^2 + 2\zeta_1\omega \cdot s + \omega^2} \cdot x(s), \quad (\text{A4})$$

where ζ_1 and ζ_2 are the damping ratios and ω the filter frequency.

The general form of the switching function used by the controller is given by the equation

$$\sigma(x_0, x_1, x) = \begin{cases} 0 & \text{for } x < x_0 \\ a_3x^3 + a_2x^2 + a_1x + a_0 & \text{for } x_0 \leq x < x_1 \\ 1 & \text{for } x \geq x_1, \end{cases} \quad (\text{A5})$$

where the coefficients take following values:

$$\begin{aligned} a_3 &= \frac{2}{(x_0 - x_1)^3} \\ a_2 &= \frac{-3(x_0 + x_1)}{(x_0 - x_1)^3} \\ a_1 &= \frac{6x_0x_1}{(x_0 - x_1)^3} \\ a_0 &= \frac{(x_0 - 3x_1)x_0^2}{(x_0 - x_1)^3}. \end{aligned} \quad (\text{A6})$$

Appendix B. Abbreviations

Table A1 lists the abbreviations used in this work.

Table A1. Abbreviations used in this work.

Abbreviation	Explanation
BEM	Blade element momentum
BRBM	Blade root bending moment
CPC	Collective pitch control
DEL	Damage equivalent load
DLC	Design load case
DTU	Technical University of Denmark
IPC	Individual pitch control
LLFVW	Lifting line free vortex wake
NREL	National renewable energies laboratory
NTM	Normal turbulence model
RWT	Reference wind turbine
TUB	Technical University of Berlin
TUBCon	TUB Controller

References

1. Burger, B. *Power Generation in Germany—Assesment of 2017*; Technical Report; Fraunhofer Institute for Solar Energy Systems ISE: Freiburg, Germany, 2018.
2. Jamieson, P. *Innovation in Wind Turbine Design*, 2nd ed.; John Wiley & Sons Ltd.: West Sussex, UK, 2018; p. 416.
3. Bossanyi, E.A. Individual Blade Pitch Control for Load Reduction. *Wind Energy* **2003**, *6*, 119–128. [[CrossRef](#)]
4. Larsen, T.J.; Madsen, H.A.; Thomsen, K. Active Load Reduction using individual Pitch, based on local Blade Flow Measurements. *Wind Energy* **2005**, *8*, 67–80. [[CrossRef](#)]
5. Jones, B.L.; Lio, W.H.; Rossiter, J.A. Overcoming fundamental limitations of wind turbine individual blade pitch control with inflow sensors. *Wind Energy* **2018**, *21*, 922–936. [[CrossRef](#)]

6. Wilson, D.; Berg, D.; Resor, B.; Barone, M.; Berg, J. Combined Individual Pitch Control and Active Aerodynamic Load Controller Investigation for the 5MW Upwind Turbine. In Proceedings of the AWEA Wind Power Conference & Exhibition, Chicago, IL, USA, 4–7 May 2009; pp. 1–12.
7. Lackner, M.; van Kuik, G.A.M. A Comparison of Smart Rotor Control Approaches using Trailing Edge Flaps and individual Pitch Control. *Wind Energy* **2010**, *13*, 117–134. [[CrossRef](#)]
8. Plumley, C.; Graham, M.; Leithead, W.; Bossanyi, E.A.; Jamieson, P. Supplementing Wind Turbine Pitch Control with a Trailing Edge Flap Smart Rotor. In Proceedings of the 3rd Renewable Power Generation Conference (RPG 2014), Naples, Italy, 24–25 September 2014; pp. 1–6. [[CrossRef](#)]
9. Jost, E.; Barlas, T.; Riziotis, V.; Navalkar, S.T. INNWIND Deliverable 2.32: Validation of New Control Concepts by Advanced Fluid-Structure Interaction Tools. Technical Report. INNWIND.eu. 2015. Available online: http://www.innwind.eu/-/media/Sites/innwind/Publications/Deliverables/2015-09-02_INNWIND_EU_D2_3_2_final2.ashx?la=da&hash=B357FF565D1B203869C1016F06D9CCE191CAD434 (accessed on 5 June 2020).
10. Chen, Z.; Stol, K.; Mace, B. System Identification and Controller Design for individual Pitch and Trailing Edge Flap Control on upscaled Wind Turbines. *Wind Energy* **2016**, *19*, 1073–1088. [[CrossRef](#)]
11. Navalkar, S.T.; Van Wingerden, J.W.; Van Solingen, E.; Oomen, T.; Pasterkamp, E.; Van Kuik, G.A.M. Subspace Predictive Repetitive Control to Mitigate Periodic Loads on Large Scale Wind Turbines. *Mechatronics* **2014**, *24*, 916–925. [[CrossRef](#)]
12. Asgharnia, A.; Jamali, A.; Shahnazi, R.; Maheri, A. Load mitigation of a class of 5-MW wind turbine with RBF neural network based fractional-order PID controller. *ISA Trans.* **2020**, *96*, 272–286. [[CrossRef](#)] [[PubMed](#)]
13. Jonkman, J.; Butterfield, S.; Musial, W.; Scott, G. *Definition of a 5-MW Reference Wind Turbine for Offshore System Development*; Technical Report TP-500-38060; NREL: Golden, CO, USA, 2009.
14. Bossanyi, E.A. The Design of closed loop controllers for wind turbines. *Wind Energy* **2000**, *3*, 149–163. [[CrossRef](#)]
15. Hansen, M.H.; Hansen, A.; Larsen, T.J.; Oye, S.; Sorensen, P.; Fuglsang, P. *Control Design for a Pitch-Regulated, Variable Speed Wind Turbine*; Technical Report R1500; Risø National Laboratory: Roskilde, Denmark, 2005.
16. Hansen, M.H.; Henriksen, L.C.; Hartvig, M.; Christian, L. *Basic DTU Wind Energy Controller*; Technical Report E-0028; DTU Wind Energy: Roskilde, Denmark, 2013.
17. Mulders, S.P.; Van Wingerden, J.W. Delft Research Controller: An open-source and community-driven wind turbine baseline controller. *J. Phys. Conf. Ser.* **2018**, *1037*, 032009. [[CrossRef](#)]
18. Abbas, N.J.; Wright, A.; Pao, L. An Update to the National Renewable Energy Laboratory Baseline Wind Turbine Controller. *J. Phys. Conf. Ser.* **2020**, *1452*, 012002. [[CrossRef](#)]
19. Meng, F.; Lio, W.H.; Barlas, T. DTUWEC: An open-source DTU Wind Energy Controller with advanced industrial features. *J. Phys. Conf. Ser.* **2020**, *1618*, 022009. [[CrossRef](#)]
20. Burton, T.; Jenkins, N.; Sharpe, D.; Bossanyi, E.A. *Wind Energy Handbook*, 2nd ed.; John Wiley & Sons Ltd.: West Sussex, UK, 2011; pp. 1–742.
21. *IEC 61400-1: Wind Turbines—Part 1: Design Requirements*; International Electrotechnical Commission: Geneva, Switzerland, 2005.
22. Madsen, H.A.; Riziotis, V.; Zahle, F.; Hansen, M.O.L.; Snel, H.; Larsen, T.J.; Politis, E.; Rasmussen, F. Blade Element Momentum Modeling of Inflow with Shear in Comparison with Advanced Model Results. *Wind Energy* **2012**, *15*, 63–81. [[CrossRef](#)]
23. Hauptmann, S.; Bülk, M.; Schön, L.; Erbslöh, S.; Boorsma, K.; Grasso, F.; Kühn, M.; Cheng, P.W. Comparison of the Lifting-Line Free Vortex Wake Method and the Blade-Element-Momentum Theory regarding the simulated Loads of Multi-MW Wind Turbines. *J. Phys. Conf. Ser.* **2014**, *555*, 012050. [[CrossRef](#)]
24. Boorsma, K.; Hartvelt, M.; Orsi, L. Application of the Lifting Line Vortex Wake Method to Dynamic Load Case Simulations. *J. Phys. Conf. Ser.* **2016**, *753*, 022030. [[CrossRef](#)]
25. Madsen, H.A.; Larsen, T.J.; Pirrung, G.R.; Li, A.; Zahle, F. Implementation of the Blade Element Momentum Model on a Polar Grid and its Aeroelastic Load Impact. *Wind Energy Sci.* **2020**, *5*, 1–27. [[CrossRef](#)]
26. Perez-Becker, S.; Saverin, J.; Marten, D.; Alber, J.; Pechlivanoglou, G.; Paschereit, C.O. Investigations on the Fatigue Load Reduction Potential of Advanced Control Strategies for Multi-MW Wind Turbines using a Free Vortex Wake Model. In Proceedings of the ASME Turbo Expo 2018, Oslo, Norway, 11–15 June 2018; p. V009T48A008. [[CrossRef](#)]
27. Perez-Becker, S.; Papi, F.; Saverin, J.; Marten, D.; Bianchini, A.; Paschereit, C.O. Is the Blade Element Momentum theory overestimating wind turbine loads?—An aeroelastic comparison between OpenFAST’s AeroDyn and QBlade’s Lifting-Line Free Vortex Wake method. *Wind Energy Sci.* **2020**, *5*, 721–743. [[CrossRef](#)]
28. Marten, D.; Lennie, M.; Pechlivanoglou, G.; Nayeri, C.N.; Paschereit, C.O. Implementation, Optimization and Validation of a Nonlinear Lifting Line-Free Vortex Wake Module within the Wind Turbine Simulation Code QBlade. *ASME J. Eng. Gas Turbines Power* **2015**, *138*, 072601. [[CrossRef](#)]
29. Bak, C.; Zahle, F.; Bitsche, R.; Kim, T.; Yde, A.; Henriksen, L.C.; Andersen, P.B.; Natarajan, A.; Hansen, M.H. *Design and Performance of a 10 MW Wind Turbine*; Technical Report I-0092; DTU Wind Energy: Roskilde, Denmark, 2013.
30. Bergami, L.; Gaunaa, M. *ATEFlap Aerodynamic Model, a Dynamic Stall Model Including the Effects of Trailing Edge Flap Deflection*; Technical Report Risø-R-1792; DTU Wind Energy: Roskilde, Denmark, 2012.
31. Wendler, J.; Marten, D.; Pechlivanoglou, G.; Nayeri, C.N.; Paschereit, C.O. An Unsteady Aerodynamics Model for Lifting Line Free Vortex Wake Simulations of HAWT and VAWT in QBlade. In Proceedings of the ASME Turbo Expo: Turbine Technical Conference and Exposition GT2016, Orlando, FL, USA, 13–17 June 2016; p. V009T46A011. [[CrossRef](#)]

32. Tasora, A.; Serban, R.; Mazhar, H.; Pazouki, A.; Melanz, D.; Fleischmann, J.; Taylor, M.; Sugiyama, H.; Negrut, D. Chrono: An Open Source Multi-Physics Dynamics Engine. In Proceedings of the International Conference on High Performance Computing in Science and Engineering, Salt Lake City, UT, USA, 13 November 2016; pp. 19–49. [[CrossRef](#)]
33. Bossanyi, E.A. Further Load Reductions with Individual Pitch Control. *Wind Energy* **2005**, *8*, 481–485. [[CrossRef](#)]
34. Plumley, C.; Leithead, W.; Jamieson, P.; Bossanyi, E.A.; Graham, M. Comparison of individual Pitch and Smart Rotor Control Strategies for Load Reduction. *J. Phys. Conf. Ser.* **2014**, *524*, 012054. [[CrossRef](#)]
35. Plumley, C. The Smart Rotor Wind Turbine. Ph.D. Thesis, University of Strathclyde: Glasgow, UK, 2015.
36. Bergami, L.; Gaunaa, M. Analysis of Aeroelastic Loads and their Contributions to Fatigue Damage. *J. Phys. Conf. Ser.* **2014**, *555*, 012007. [[CrossRef](#)]
37. Iribas, M.; Hansen, M.H.; Mahmood, M.; Tibaldi, C.; Natarajan, A.; Bossanyi, E.; Stock, A.; Jamieson, P.; Leithead, W.; Schlipf, D. INNWIND Deliverable 1.42: Methodology for Feed-Forward Control Strategies Using Nacelle or Blade Based Sensors and Distributed Control. Technical Report. INNWIND.eu. 2015. Available online: <http://www.innwind.eu/publications/deliverable-reports> (access on 5 June 2020).
38. Borg, M.; Mirzaei, M.; Bredmose, H. *LIFES50+ Deliverable D1.2: Wind Turbine Models for the Design*; Technical Report E-101; DTU Wind Energy: Roskilde, Denmark, 2015.
39. Hayman, G.J. *MLife Theory Manual for Version 1.00*; Technical Report; NREL: Golden, CO, USA, 2012.
40. Buhl, M.L.J. Crunch Software. Available online: <https://nwtc.nrel.gov/Crunch> (accessed on 27 January 2018).
41. Bergami, L. Adaptive Trailing Edge Flaps for Load Reduction. In Proceedings of the 7th PhD Seminar on Wind Energy in Europe, Delft, The Netherlands, 27–28 October 2011.