Lidar-assisted model predictive control of wind turbine fatigue via online rainflow-counting considering stress history

Stefan Loew and Carlo L. Bottasso

Wind Energy Institute, Technical University of Munich, 85748 Garching b. München, Germany

Correspondence: Carlo L. Bottasso (carlo.bottasso@tum.de)

Abstract. The formulation of Parametric Online Rainflow Counting implements the standard fatigue estimation process and a stress history in the cost function of a Model Predictive Controller. The formulation is tested in realistic simulation scenarios where the states are estimated by a Moving Horizon Estimator and the wind is predicted by a lidar simulator. The tuning procedure for the controller toolchain is carefully explained. In comparison to a conventional MPC in a turbulent wind setting, the novel formulation is especially superior with low lidar quality, benefits more from the availability of a wind prediction, and exhibits a more robust performance with shorter prediction horizons. A simulation excerpt with the novel formulation provides deeper insight into the update of the stress history and the fatigue cost parameters. Finally, in a deterministic gust setting, both the conventional and the novel MPC – despite their completely different fatigue cost – exhibit similar pitch behavior and tower oscillations.

1 Introduction

Fatigue is damage of a material caused by cyclic application of mechanical stress. For wind turbines, fatigue has a large impact on lifetime e.g. of tower, blades and drivetrain, and is a main design driver. Model Predictive Controllers (MPC) enable optimal control of turbines by utilizing predictions of the incoming wind by a *light detection and ranging* (lidar) device (Bottasso et al., 2014; Schlipf et al., 2013). Based on these input predictions, stress time series at crucial spots in the turbine structure can be predicted. Rainflow-counting (RFC) is the standard method for the decomposition of stress time series for fatigue estimation. Until recently, RFC could not be implemented in MPC (Barradas-Berglind and Wisniewski, 2016) and could only be used for post-processing of measured and simulated data. In Loew et al. (2020a), an MPC formulation was presented that externalizes the RFC evaluation and includes its results back into the MPC via time-varying parameters. Therefore, this formulation is referred to as *Parametric Online Rainflow Counting (PORFC)*. PORFC allows for the direct and rigorous incorporation of fatigue in the cost function or constraints of MPC.

In PORFC, fatigue is calculated based on stress information from the prediction horizon of the MPC, which is in the order of a few seconds. However, fatigue is a long-term effect where stress cycles are usually defined on much longer time spans. Therefore, in Loew et al. (2020b) PORFC was combined with a systematic incorporation of historic stress samples ("residue"). In the same work, this formulation was simulated in an idealized setting where only a few degrees of freedom (DOF) in the plant turbine model were activated, and where full information about the incoming wind and the turbine states was assumed.

The main goal of the present work is to thoroughly assess the formulation in a more realistic simulation scenario. Particularly, a mismatch is introduced between the MPC-internal and the plant models, a Moving Horizon Estimator provides initial state estimates for the MPC, and a lidar simulator is utilized to generate a realistically imperfect wind estimate. The assessment is performed in several turbulent as well as deterministic gust scenarios.

This paper is organized as follows. In Sect. 2, the phenomenon of fatigue and cycle identification are reviewed. This analysis is the basis for an application-focused description of PORFC in Sect. 3. In Sect. 4, a Moving Horizon Estimator is formulated. In Sect. 5, the controller toolchain and the tuning of each of its elements are presented. Finally, PORFC is compared to a conventional MPC and to a conventional PID controller in the above mentioned simulation scenarios.

2 Review of fatigue estimation

In the following, fatigue is defined, cycle identification is explained, and the concept of *residue* is presented.

2.1 Definition of fatigue

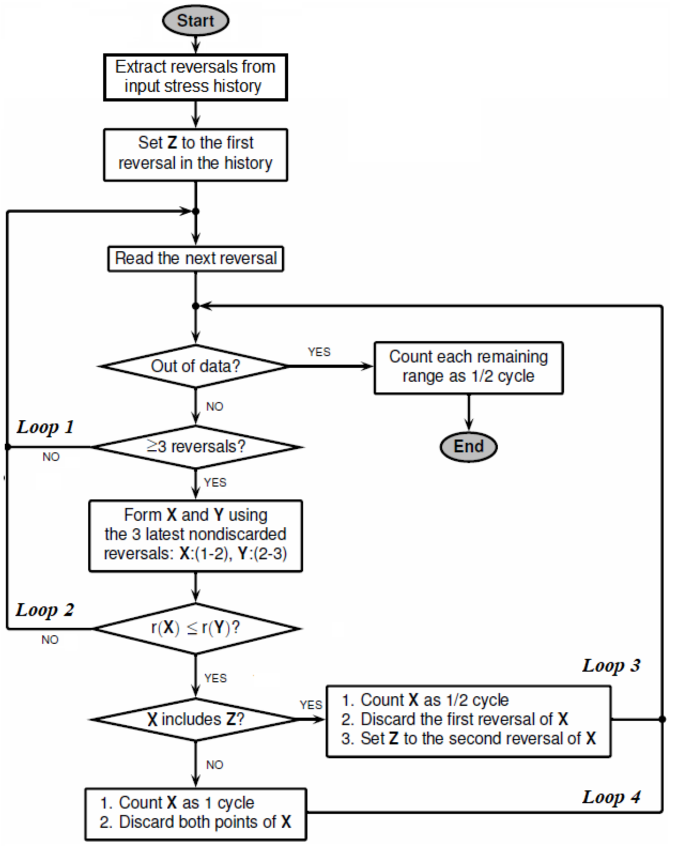
In the following, the phenomenon of fatigue is defined for conditions and assumptions that apply to the wind energy domain: namely mechanical fatigue, normal ambient temperatures, neglection of irreversible strain effects and invariance with respect to time. In this setting, fatigue is damage of a material caused by cyclic application of mechanical stress. Without loss of information, the fatigue impact of a given stress-trajectory can be analyzed solely based on its extrema or "reversals". This implies that the shape and contained frequencies of the original continuous stress trajectory are considered to be irrelevant for fatigue estimation (Barradas-Berglind et al., 2015). Therefore, the fatigue impact of a reversal sequence is fully determined by its contained individual stress cycles. Each stress cycle can be represented by a cosine function. A stress trajectory typically contains full cycles, which are cosines of a full period, and half cycles, which are cosines of only a half period. Half cycles therefore represent either a rising or falling transient. Instead of storing three (full cycle) or two (half cycle) stress samples, it is common to store two stress samples and a weight, which is valued $w_c = 1$ (full cycle) or $w_c = 0.5$ (half cycle). The two stress samples can be the cycle stress maximum and minimum, or the stress amplitude $\sigma_{\mathrm{a},c}$ and mean $\sigma_{\mathrm{m},c}$. Instead of stress amplitude, stress range $\sigma_{\mathrm{r},c} = 2\sigma_{\mathrm{a},c}$ is frequently used as well.

Typically, fatigue impact of a stress cycle mainly correlates with its stress amplitude: a positive stress mean increases whereas a negative stress mean decreases fatigue impact. Quantitatively, this *mean stress effect* is expressed by the Goodman equation (Haibach, 2006) (p. 184), which leads to the equivalent stress $\sigma_{\mathrm{eq},c}$. Consequently, equivalent stress is used to calculate the number of cycles to failure

$$N_c = f_{\mathrm{SN}}^{-1}(\sigma_{\mathrm{eq},c}) \tag{1}$$

via the inverse S-N or *"Woehler"* curve, which typically has a piecewise definition over the stress-axis. Fatigue damage of a given stress cycle

$$D_{\mathrm{fatigue,c}} = 1/N_c \tag{2}$$



Start
Extract reversals from input stress history
Set Z to the first reversal in the history
Read the next reversal
Out of data?
YES
Count each remaining range as 1/2 cycle
End
NO
Loop 1
≥3 reversals?
NO
YES
Form X and Y using the 3 latest nondiscarded reversals: X:(1-2), Y:(2-3)
Loop 2
r(X) ≤ r(Y)?
NO
YES
Loop 3
YES
X includes Z?
1. Count X as 1/2 cycle
2. Discard the first reversal of X
3. Set Z to the second reversal of X
NO
Loop 4
1. Count X as 1 cycle
2. Discard both points of X

Figure 1. Flowchart of the MATLAB-implementation `rainflow()` of the *Three-point algorithm* (simplified from The MathWorks Inc. (2018)). Stress extrema are called "reversals". The range $r(\boldsymbol{X}) = |X(2) - X(1)|$ of a stress value pair $\boldsymbol{X}$ is the absolute value of the difference between both stresses.

is obtained by the reciprocal of the number of cycles to failure. Total damage of the given stress trajectory is obtained by linear accumulation $D_{\text{fatigue}} = \sum_c D_{\text{fatigue,c}}$ of damages of individual stress cycles according to the *Miner-Palmgren-Rule* (Miner, 1945).

2.2 Cycle identification via the Rainflow algorithm

Cycle identification is straightforward if, e.g., a simple sinusoid is analysed. There, amplitudes, mean values and number of cycles are obvious. However, realistic stress trajectories usually are highly complex and contain stress cycles that can be nested *("nested cycles")*. Additionally, half and full cycles can be present, as stated above. The most widely accepted algorithm for cycle identification from complex trajectories is the Rainflow(-counting) algorithm (RFC) (ASTM International, 1985). A flowchart of the Rainflow algorithm is displayed in Fig. 1.

At the beginning of the algorithm, RFC receives as input a stress trajectory and extracts its reversals (extrema). Throughout the algorithm, reversals are read consecutively from left to right. Each new reversal is stored in an operational memory. From this memory, cycles are identified based on a triplet of reversals. The Rainflow algorithm contains four main loops. *Loop 1* initiates the reading of a new reversal sample, if less than three reversals are in the operational memory. *Loop 2* initiates the reading of a new reversal if, based on the current operational memory, no cycle could be identified. *Loop 3* and *Loop 4* initiate the subsequent check for a cycle in the current operational memory and are triggered after identification of a half or full cycle, respectively. A more comprehensive explanation of the algorithm can be found in The MathWorks Inc. (2018).

As shown above, the Rainflow algorithm contains algorithmic branches and loops. Thus, a crucial property of the Rainflow algorithm is its discontinuous output behavior. Furthermore, the number N_c of identified cycles is not known before execution, but bounded by the number of extrema.

The characteristics of the identified cycles that are output by RFC for each cycle c are: stress range $\sigma_{\mathrm{r},c}$ [Pa], stress mean $\sigma_{\mathrm{m},c}$ [Pa], sample index of cycle start $k_{\mathrm{start},c}$ [-], sample index of cycle end $k_{\mathrm{end},c}$ [-], and cycle weight $w_c[-]$. In the present work, these characteristics will be used in a converted form of stress amplitude $\sigma_{\mathrm{a},c}$ [Pa], stress mean $\sigma_{\mathrm{m},c}$ [Pa], sample index of cycle maximum $k_{\mathrm{max},c}$ [-], sample index of cycle minimum $k_{\mathrm{min},c}$ [-], and cycle weight $w_c[-]$.

2.3 Batchwise cycle identification and residue

As shown in Loew et al. (2020b), wind turbine stress trajectories can contain long-term cycles. Thus, the Rainflow analysis has to be carried out over the entire length of an available stress trajectory. For offline purposes, this mode is perfectly adequate. However, for online monitoring and control, a complete Rainflow analysis for each newly measured stress sample is computationally infeasible. As a solution, Heinrich et al. (2019) showed that Rainflow analysis can be performed batchwise if a so-called *residue* is used for carrying along the half-cycle stress samples. Residue, therefore, denotes a set of stress samples that occured in the past and have not formed full cycles as yet.

Depending on the stress signal, a high number of samples can be accumulated in the residue. The maximum possible length of the residue vector results from diverging and converging stress time series, because they generate a large number of half cycles (Köhler et al., 2012). However, long-term diverging series are unrealistic, because unstable machine behavior typically is counteracted by the controller or an emergency shutdown. Long-term converging series are irrelevant, since very low-amplitude cycles can be discarded without significant errors in fatigue estimation. To conclude, the length of the residue vector is finite and remained well below 100 in practical tests (Loew and Obradovic, 2020).

3 Fatigue in Model Predictive Control of wind turbines

Wind turbine fatigue is usually implemented in MPC within the cost function. Common cost types in MPC are *Stage cost* and *Terminal cost*. Stage costs comprise a summation of state samples or a time integral of state trajectories over the prediction horizon, and are preferred for the present application. Terminal costs are defined as a function of the sole state samples at the end of the prediction horizon (Grüne and Pannek, 2017). Alternatively, fatigue can also be used as a constraint, for example to express a desired lifetime goal.

3.1 Indirect fatigue metrics in MPC

Several approaches reported in the literature involve indirect fatigue metrics (Barradas-Berglind and Wisniewski, 2016; Gros and Schild, 2017; Evans et al., 2015). However, indirect fatigue metrics have two main disadvantages:

- Instead of actual damage, only a damage-related value is obtained and optimized.
- Indirect fatigue terms have different units from harvested energy. Thus, weighting both terms in the cost function is not straightforward.

When considering tower fatigue, the most common approach involves the quadratic penalization of tower tip deflection rate $\dot{d}_\mathrm{T}$. This also can be interpreted as a penalization of kinetic energy of the lumped tower mass m_T, averaged over the prediction horizon T_horiz. In the present work, therefore, the stage cost

$$J_{\mathrm{fatigue,TTVP}} = \int\limits_{t_0}^{t_\mathrm{end}} \frac{1}{2T_\mathrm{horiz}P_\mathrm{g,max}} m_\mathrm{T}\dot{d}_\mathrm{T}^2 \mathrm{d}t \tag{3}$$

is used for comparison and referred to as *Tower Tip Velocity Penalization (TTVP)*. An additional division by rated power $P_\mathrm{g,max}$ is used for scaling the cost, which is beneficial for optimization.

3.2 Direct fatigue metrics in MPC

In contrast to indirect fatigue metrics, direct fatigue metrics return actual damage. As shown in Sect. 2, direct fatigue estimation involves the Rainflow algorithm. Implementation of RFC within a gradient-based optimization seemed impossible until now due to the following obstacles:

- RFC is a function of all stress samples. Therefore, neither the concept of stage nor of terminal cost applies.
- RFC contains branches. Therefore, it exhibits discontinuous outputs and is not continuously differentiable.
- RFC contains "while" loops, which lead to a changing function execution structure depending on the stress input.

Thus, in all known references, the Rainflow algorithm is approximated to some extent. In Sanchez et al. (2015), a version of *Simple Range Counting* is applied, which is standardized in ASTM International (1985). In Barradas-Berglind et al. (2015), hysteresis operators are used to adapt parameters of a cost function in MPC. This cost function penalizes deflection rates, comparable to TTVP. In Luna et al. (2020), damage estimation including standard RFC is performed on a large number of stress time series, which are used to train a surrogate Artificial Neural Network (ANN). The latter seems to be very promising in terms of correct damage estimation. However, the approach involves a high a priori effort in setting up the ANN, as well as a significantly increased computational load in the MPC (Luna et al., 2020).

Stress history is not included in any of these approaches. In Barradas-Berglind et al. (2015), the hysteresis operators only have memory of damage evolution. Similarly, in Luna et al. (2020), only the previous fatigue rate output of the ANN is memorized until the next evaluation.

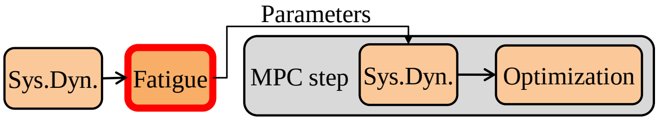


Figure 2. Externalization of fatigue estimation (Rainflow algorithm) from the MPC.

3.3 Parametric Online Rainflow Counting - Concept

The above mentioned obstacles for a direct implementation of RFC in MPC are overcome by the method of *Parametric Online Rainflow Counting (PORFC)*. In PORFC, all discontinuous parts of the fatigue estimation procedure are carried out before each execution of the MPC algorithm, as shown in Fig. 2. Additionally, the stress history is incorporated via a residue, which is inspired by the batchwise cycle identification in Sect. 2.3. The algorithmic workflow is as follows:

- **Simulation:** The reduced wind turbine model is simulated over the prediction horizon using the current measured states as initial values to produce a stress prediction, as visualized in Fig. 3b.

- **Merge:** The residue (see Fig. 3a) is merged with the stress prediction.

- **Rainflow:** The Rainflow algorithm is used to identify stress cycles over this merged trajectory. Consequently, it is assumed that the structure of identified cycles does not change within the next optimization. The term "structure" denotes here positions ($k_{\mathrm{min,c}}$, $k_{\mathrm{max,c}}$) and weights (w_c) of cycles. As shown in Fig. 3b, this assumption implies that the controllable extrema in the prediction horizon only can be shifted vertically (i.e., in the values but not in their positions) by the optimization.

- **Residue update:** Stress cycles can be composed by stress samples only from residue or prediction, or by a combination of both ("mixed cycle"). However, only the samples within the prediction horizon can be controlled by the optimization. Particularly the measured initial value at prediction step 0 cannot be controlled and, therefore, is added to the residue. If a full cycle is detected entirely within the *residue*, both contributing values are discarded from the residue. The reason for this is that also in the future they will never anymore form a cycle with a sample from the prediction and, therefore, are irrelevant for the MPC.

- **Time-varying parameters:** Information from cycle identification is used to fill vectors of time-varying parameters, which are forwarded to the cost function of the MPC. Details on this step are provided in Sect. 3.4.

- **Optimization/MPC:** In the cost function of the MPC, the parameters are used to time-continuously calculate fatigue cost over the horizon and accumulate it via integration. Finally, the optimization problem is solved and the resulting control variables are applied to the wind turbine plant.

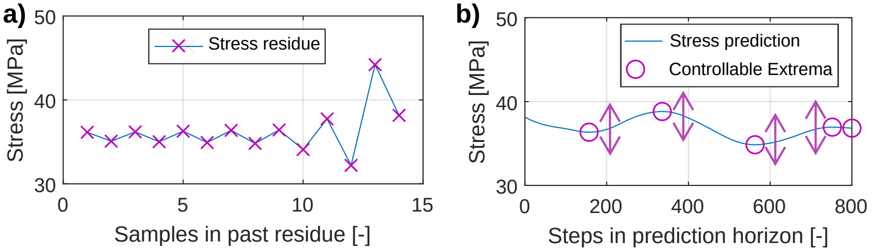


Figure 3. Stress residue from the past (a). Stress prediction into the future (b).

3.4 Parametric Online Rainflow Counting - Time-varying parameters and cost function

3.4.1 Distribution of damage over time

Since information from cycle identification is forwarded to the MPC via parameters, which are varying over the prediction horizon, the total fatigue damage has to be distributed over the prediction horizon, as visualized in Fig. 4b. Therefore, the damage of each stress cycle is split into two halves, which are allocated to the two contributing stress samples. For example, cycle 4 is formed by samples $k = 4$ and $k = 8$. Their fatigue cost terms therefore are allocated to these samples, as shown by the blocks in Fig. 4b. This example also shows an important property of the Rainflow algorithm, which identifies cycle 4 even though it is interrupted by the nested cycle 2, as shown in Fig. 4a. If, for a given stress sample, the complementary stress sample is not controllable (i.e. lies in the residue), all damage is allocated to the given sample. Here, this is the case for cycles 1 and 3, where all damage is allocated to sample $k = 2$ and $k = 4$, respectively.

3.4.2 Setup of the time-varying parameters

Figure 4a visualizes the generation of the time-varying parameters. Since each stress extremum belongs to one or two stress cycles (Shi et al., 2018), one or two stress references are set per extremum. These stress references are considered as optimization- or tracking-references for the current MPC-step. If both stress samples of a cycle lie in the prediction, mean stresses (M2, M4) become the stress references. If the complementary stress sample of a cycle lies in the uncontrollable residue ("mixed cycle"), this complementary stress value (C1, C3) becomes the stress reference for the considered sample in the controllable prediction. However, in many cases, a mixed cycle is crossing the level of the initial stress $\sigma(t_0)$. In this case, the best possible tracking-reference is this initial stress value itself (R1, R3), since zero oscillation in the prediction corresponds to zero fatigue cost. A more detailed derivation and explanation can be found in Loew et al. (2020a).

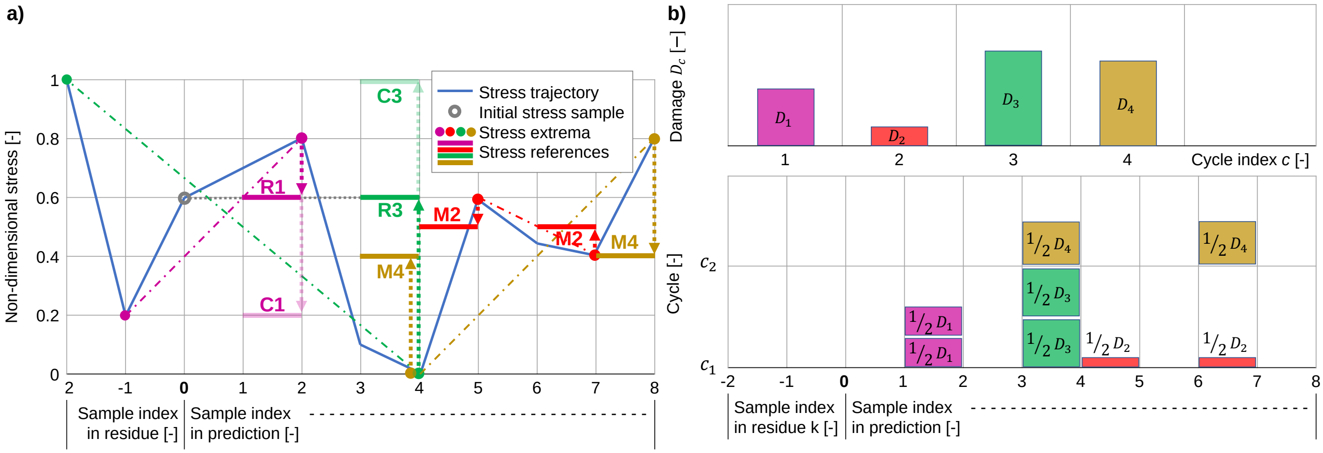


Figure 4. Stress trajectory (blue), its initial value at t_0 (grey circle), its extrema (colored dots), sequence of samples that form a cycle (dash-dotted), generated time-varying reference stresses (solid purple, red, green, yellow), and optimization goals (dotted arrows) for PORFC (a). Corresponding distribution of damage over the prediction horizon (b). Both figures are modified from Anand (2020).

3.4.3 Cost function

The fatigue cost function is defined by an integral over two cost terms, each one representing one potential cycle contribution of a stress sample, i.e.:

$$J_{\mathrm{fatigue,PORFC}}\left(\sigma,\bar{\boldsymbol{p}}\right) = \frac{1}{T_{\mathrm{cntrl}}}\int_{t_0}^{t_{\mathrm{end}}}\left(J_{\mathrm{fatigue},c}(\sigma(t),\hat{\bar{\sigma}}_{\mathrm{ref},c1}(t),\hat{\bar{w}}_{c1}(t)) + J_{\mathrm{fatigue},c}(\sigma(t),\hat{\bar{\sigma}}_{\mathrm{ref},c2}(t),\hat{\bar{w}}_{c2}(t))\right)\,\mathrm{d}t \quad [€]. \tag{4}$$

The notation $\hat{(\cdot)}$ means fixed for one MPC-step, while $\bar{(\cdot)}$ means sampled on the control intervals of the prediction horizon. The cost terms are "switched on" by nonzero cycle weights $\hat{\bar{w}}_{c1/2}(t)$. Reference stresses $\hat{\bar{\sigma}}_{\mathrm{ref},c1/2}(t)$ and cycle weights $\hat{\bar{w}}_{c1/2}(t)$ are collected in the parameter vector

$$\bar{\boldsymbol{p}} = \left(\hat{\bar{\sigma}}_{\mathrm{ref},c1}, \hat{\bar{\sigma}}_{\mathrm{ref},c2}, \hat{\bar{w}}_{c1}, \hat{\bar{w}}_{c2}\right), \tag{5}$$

which is defined as piecewise constant over the control intervals of the prediction horizon. The cost of individual cycles is defined by

$$J_{\mathrm{fatigue},c} = \hat{\bar{w}}_{c1/2}(t)\; a_{\mathrm{m}}\; |\sigma(t) - \hat{\bar{\sigma}}_{\mathrm{ref},c1/2}(t)|^{m}, \tag{6}$$

where the fatigue coefficient a_{m} and the fatigue exponent m are derived from the damage curve of Eq. (2).

3.5 Optimization problem for TTVP and PORFC

The rigourous inclusion of fatigue into an MPC formulation described up to now is completely general, and can be used to formulate different cost functions. The same formulation can be readily adjusted to include fatigue damage as an MPC constraint.

To exemplify the use of PORFC to a practical case, here the following economic optimization problem is considered

$$\min_{\bar{\boldsymbol{u}},\bar{\boldsymbol{s}}} \left(-\alpha_{\mathrm{revenue}} J_{\mathrm{revenue}} + \alpha_{\mathrm{fatigue}} J_{\mathrm{fatigue}} + \int\limits_{t_0}^{t_{\mathrm{end}}} \left(10^{-1} \dot{\beta}_{\mathrm{b}}^2 + 10^{-2} \dot{T}_{\mathrm{g}}^2 + 10^7\, \bar{s}_{\omega}^2 + 10^7\, \bar{s}_P^2 \right) \mathrm{d}t \right). \tag{7}$$

The problem seeks the maximization of the revenue J_{revenue} and the minimization of the fatigue J_{fatigue}, which is represented by Eq. (3) for TTVP or Eq. (4) for PORFC. The constants $\alpha_{\mathrm{revenue}}$ and $\alpha_{\mathrm{fatigue}}$ are weighting factors. Instead of generated electrical energy, harvested aerodynamic energy J_{revenue} is maximized to avoid a greedy extraction of rotor kinetic energy by the MPC (*turnpike effect*), as suggested by Gros and Schild (2017). Furthermore, pitch rate $\dot{\beta}_{\mathrm{b}}$, torque rate $\dot{T}_{\mathrm{g}}$, and slack variables for rotational speed $\bar{s}_{\omega}$ and generator power $\bar{s}_P$ are penalized (see their use in the constraints below). The optimization variables are the demanded pitch angle and torque rate $\bar{\boldsymbol{u}} = (\bar{\boldsymbol{\beta}}_{\mathrm{b,d}}, \dot{\bar{\boldsymbol{T}}}_{\mathrm{g,d}})$, and the slack variables $\bar{\boldsymbol{s}} = (\bar{\boldsymbol{s}}_{\omega}, \bar{\boldsymbol{s}}_P)$. For both TTVP and PORFC, revenue is weighted by the current electricity price $\alpha_{\mathrm{revenue}} = p_{\mathrm{elec}}$ [€W^{-1} s^{-1}] to match the monetary nature of Eq. (4). The fatigue weight $\alpha_{\mathrm{fatigue}}$ remains free and will be determined later in this work.

It should be noted that the balance of revenue and tower fatigue, despite being common in wind turbine MPC research (Gros, 2013; Evans et al., 2015; Luna et al., 2020), does not fully reflect the true economic goals of wind turbine operators, nor does it capture the complex interrelations among power capture, damage to the various turbine components and its effects on operation and maintenance costs, on lifetime, on actuator duty cycle, and others. Therefore, the novelty of the present contribution is on how fatigue is treated in a modern control framework, not on the specific formulation of the cost function. A more realistic industrial application should embed damage into a more complex business-oriented scenario-dependent cost function or constraint. This aspect of the problem is extremely relevant and very interesting, but it is considered as out of scope for the present work.

The optimization problem is subject to:

- The system dynamics of a reduced turbine model $\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{d}(t))$, whose six states

  $$\boldsymbol{x}(t) = (\omega_{\mathrm{r}}(t), d_{\mathrm{T}}(t), \dot{d}_{\mathrm{T}}(t), \beta_{\mathrm{b}}(t), \dot{\beta}_{\mathrm{b}}(t), T_{\mathrm{g}}(t))^T \tag{8}$$

  are rotational speed of the rotor noted ω_{r}, tower tip deflection d_{T}, tower tip velocity $\dot{d}_{\mathrm{T}}$, pitch angle β_{b}, pitch rate $\dot{\beta}_{\mathrm{b}}$, and generator torque T_{g}. More details about the model are given in Löw and Obradovic (2018).

- Inequality constraints over the horizon, to keep rotational speed, tower deflection (yield strength), pitch angle, pitch rate, generator torque, and generator power within their limits. In order to maintain feasibility of the optimization despite model uncertainties and temporary constraint violations, the constraints on rotational speed and generator power are augmented by slack variables, as suggested by Gros (2013).

– Box constraints on control and slack variables.

4 Moving Horizon Estimator

The MPC-internal system model only comprises the 6 states defined by Eq. (8), while the plant model in `OpenFAST` (including the actuators, but excluding the yaw mechanism) comprises 33 states (8 tower states, 6 states for each of the 3 blades, 2 states for drive-shaft torsion, 2 states for rotor rotation, 2 states for the collective blade pitch actuation, 1 state for the generator torque actuation) (Jonkman et al., 2009). Thus, the MPC-internal model is only a reduced representation of the plant model. Furthermore, both the tower deflection and velocity of the MPC-internal model cannot be measured directly on a real turbine. Only rotor speed, tower tip fore-aft acceleration $\ddot{d}_{\mathrm{T}}(t)$ and the actuator states can be measured by on-board sensors. Consequently, a state estimator is required to provide initial value estimates for the MPC-internal model based on the available measurements from the plant and the lidar system.

Kalman filters are widely used for the estimation of structural wind turbine states (Bottasso and Croce, 2009; Ritter, 2020). However, they have some disadvantages compared to the more sophisticated Moving Horizon Estimators (MHEs). First, Kalman filters are minimum-variance state estimators for linear dynamic systems with Gaussian noise; although assumptions on linearity and Gaussian noise behavior can be relaxed, MHEs are formulated as more general non-linear optimization problems over a time horizon, which represent a natural complement to the similarly general non-linear optimization-based formulations behind MPC. Second, the inclusion of constraints in state estimation problems can be important to prevent non-physical results. The inclusion of state constraints is possible in Kalman filters, but not straightforward, and non-linear constraints lead to loss of optimality of the filter and may generate different results, depending on the formulation (Simon, 2010). In contrast, state constraints can be explicitly and readily set in an MHE (Rawlings et al., 2017). Although state constraints are not employed in the estimator used in the present work, this feature may become relevant in future research. Third, Kalman filters are one-step recursive estimation methods, and thus have to start operation with only one measurement time sample. In case of large initial state errors, this can lead to inaccurate estimation and possibly to the divergence of the filter (Rawlings et al., 2017). MHEs are less vulnerable to this danger since, right from the start, they take into account an estimation horizon comprising numerous measurement samples. An advantage of the recursive nature of Kalman filters is their significantly lower computational effort compared to MHEs. However, in the present context, the computational effort of an MHE is low enough if suboptimal optimization methods such as the “Real-time iteration” (Gros et al., 2013) are utilized. Because of these advantages of MHEs over Kalman filters, the former are chosen for the present work.

4.1 Formulation of the MHE

The cost function of the MHE

$$\min_{\bar{\boldsymbol{v}}} \int\limits_{t_0-T_{\mathrm{horiz,est}}}^{t_0} \left(\|\boldsymbol{y}_{\mathrm{est}}(t) - \boldsymbol{y}_{\mathrm{meas}}(t)\|^2_{\boldsymbol{W}_{\mathrm{meas}}} + \|\boldsymbol{x}_{\mathrm{est}}(t) - \boldsymbol{x}_{\mathrm{est,prev}}(t)\|^2_{\boldsymbol{W}_{\mathrm{prev}}} + \|\boldsymbol{v}(t)\|^2_{\boldsymbol{W}_v} \right) \mathrm{d}t \tag{9a}$$

penalizes differences of the current estimates from the measurements, differences of the current estimates from the previous estimates, as well as noise $\boldsymbol{v}(t)$ (Huang et al., 2010; Gros et al., 2013). The term $\boldsymbol{x}_{\mathrm{est,prev}}(t)$ indicates the state trajectories that were estimated at the previous MHE execution, assumed as piece-wise constant and shifted backward in time by one time step. This second term of the cost function penalizes deviations over the course of consecutive MHE steps, and has been added to obtain a smoother estimation output. The noise term $\bar{\boldsymbol{v}}$ is also assumed to be piece-wise constant. Within the vectors of estimated

$$\boldsymbol{y}_{\mathrm{est}} = \begin{pmatrix} \boldsymbol{x}_{\mathrm{est}}(t) \\ \ddot{d}_{\mathrm{T,est}}(t) \end{pmatrix} \tag{9b}$$

and measured variables

$$\boldsymbol{y}_{\mathrm{meas}} = \begin{pmatrix} \boldsymbol{x}_{\mathrm{meas}}(t) \\ \ddot{d}_{\mathrm{T,meas}}(t) \end{pmatrix}, \tag{9c}$$

the states $\boldsymbol{x}$ are defined as in the reduced system given by Eq. (8). The estimated tower acceleration $\ddot{d}_{\mathrm{T,est}}(t)$ is obtained by the nonlinear output equation

$$\ddot{d}_{\mathrm{T,est}}(t) = \frac{1}{m_{\mathrm{T}}} \left(F_{\mathrm{T}}(t) - c_{\mathrm{T}} \dot{d}_{\mathrm{T,est}}(t) - k_{\mathrm{T}} d_{\mathrm{T,est}}(t) \right), \tag{9d}$$

with lumped tower mass m_{T}, damping c_{T} and stiffness k_{T}. The diagonal weighting matrices $\boldsymbol{W}_{\mathrm{meas}}$, $\boldsymbol{W}_{\mathrm{prev}}$ and $\boldsymbol{W}_{v}$ will be tuned in Sect. 5.2.4.

The optimization problem is only subject to the system dynamics

$$\dot{\boldsymbol{x}}_{\mathrm{est}} = \boldsymbol{F}(\boldsymbol{x}_{\mathrm{est}}(t), \boldsymbol{d}_{\mathrm{est}}(t)) + \boldsymbol{v}(t), \tag{9e}$$

with the additive optimization variable represented by the process noise $\boldsymbol{v}(t)$ (Huang et al., 2010). The external input

$$\boldsymbol{d}_{\mathrm{est}}(t) = (V_{\mathrm{w}}(t), \beta_{\mathrm{b,d}}(t), \dot{T}_{\mathrm{g,d}}(t))^{T} \tag{9f}$$

comprises the lidar-estimated wind speed V_{w}, as well as the pitch angle $\beta_{\mathrm{b,d}}$ and torque rate demands $\dot{T}_{\mathrm{g,d}}$, which have been set by the MPC and thus are fixed for the present MHE step. Notably, there is no equality constraint for the initial states $\boldsymbol{x}_{\mathrm{est}}(t_0 - T_{\mathrm{horiz,est}})$, which thus are freely varied by the optimization algorithm.

After the execution of the MHE, the terminal states at the end of the MHE estimation horizon become the initial states at the beginning of the MPC prediction horizon:

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_{\mathrm{est}}(t_0). \tag{10}$$

In the present controller setup, the optimized noise $\boldsymbol{v}(t)$ of the MHE is not utilized in the MPC, and its role is limited to the improvement of the quality of the estimates by taking into account process noise (which, in this context, also includes model errors).

4.2 Initialization of the MHE

The MHE requires information about the measurements over its entire estimation horizon. Therefore, the past measurements $\boldsymbol{y}_{\text{meas}}$ are buffered.

As mentioned above, the tower deflection and velocity are not measured. However, the MHE optimization benefits from meaningful measurement values as tracking reference. Therefore, a static wind-to-tower-deflection mapping is interpolated over the lidar wind estimate in order to generate a proxy tower deflection trajectory $d_{\text{T,meas}}$. Tower velocity $\dot{d}_{\text{T,meas}}$ is obtained by the numerical time-derivative of the deflection trajectory. These quantities are termed "lidar-based references" in the remainder of this work.

5 Simulation setup, tuning and results

In the following, the simulation setup is presented, each element of the controller toolchain is tuned, and the simulation results are discussed.

5.1 Simulation setup

5.1.1 Plant model

Various controller formulations are tested with the NREL 5MW onshore reference turbine (Jonkman et al., 2009) in the aeroelastic simulator `OpenFAST`. This turbine has a hub height of 110m and a rotor diameter $D = 126$m. All mechanical degrees of freedom (DOF) are activated.

5.1.2 Wind model

All turbulent results in this work are mean values of 12 simulations (each with a different seed) of 600s length in DLC 1.2 with category A turbulence. For the chosen turbine with a hub height of 110m and coastal onshore setting, a mean annual wind speed of $V_{\text{w,mean}} = 7\text{ms}^{-1}$ can be assumed (Hau, 2017). Thus, in the turbulent simulations the probability of wind speed is assumed to follow the Rayleigh distribution

$$p(V_{\text{w}}) = \frac{\pi}{2}\left(\frac{V_{\text{w}}}{V_{\text{w,mean}}^2}\right)\exp\left[-\frac{\pi}{4}\left(\frac{V_{\text{w}}}{V_{\text{w,mean}}}\right)^2\right], \tag{11}$$

as shown in Fig. 5. The Rayleigh distribution is a variant of the Weibull distribution with the simplification of having only a single parameter (Manwell et al., 2002).

5.1.3 Lidar simulator

The model of a pulsed lidar with four beams is employed. The model is implemented in the lidar simulator from `sowento GmbH`, which generates lidar wind estimates offline, and thus independently from the wind turbine simulation suite (Raach and

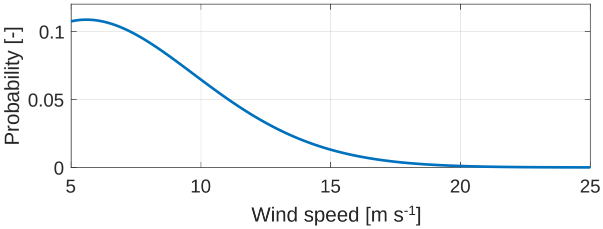


Figure 5. Rayleigh probability density function for a mean wind speed of $V_{\mathrm{w,mean}} = 7\mathrm{ms}^{-1}$.

Schlipf, 2018). Considered physical effects are: limitation to line-of-sight wind speeds, spatial averaging via a Gaussian range weighting function, discrete scanning, and "unfrozen" wind evolution. Particularly the wind evolution can be parameterized by an exponential decay constant; here, a higher value results in higher variation of the wind during its convection towards the rotor. Finally, the spatially distributed measurements are converted to rotor-effective wind speed by wind field reconstruction.

5.1.4 MHE/MPC framework

The MHE and MPC are implemented in the state-of-the-art `acados` framework (Verschueren et al., 2021), using the interior-point solver `HPIPM` for the underlying Quadratic Programs (QP).

5.1.5 Controller variants

In the following, the performance of five MPC formulations and the baseline conventional controller (CC) from NREL (Jonkman et al., 2009) are compared. The MPCs involve the conventional formulation of TTVP (see Sect. 3.1) and the novel formulation of PORFC (see Sect. 3.3). For PORFC, a fatigue exponent of $m = 2$ (see Eq. (6)) is utilized that, resulting in quadratic cost functions, is particularly suited for Quadratic Programming. This case is assessed in combination with (named PORFC-2R) and without (named PORFC-2) the use of residue. Additionally, a fatigue exponent of $m = 5$ is also tested (named PORFC-5), because this value is present at low stress amplitudes in the actual S-N-curve of the tower material. However, this parameterization in combination with residue (which would be named PORFC-5R) has not led to satisfactory results, and thus is not considered further.

5.1.6 Performance indicators

Considered performance indicators are revenue (analog to energy), fatigue cost derived from damage at tower base (based on tower CAPEX and a realistic piecewise S-N-curve), a simplified definition of profit (revenue subtracted by fatigue cost), pitch travel, and torque travel. It should be noted, that these definitions of fatigue cost and profit have various limitations:

- A change of tower damage can have further implications, e.g. on maintenance costs, which are not considered here.
- Changes in the pitch and torque utilization affect fatigue on other components, such as the blades and drive-train, an effect that is not considered here.

– The computation of the actual profit of the operator is potentially much more complex and scenario-dependent than the simple model considered here.

However, the focus of the present work is on the demonstration of the rigourous inclusion of fatigue in an MPC framework, rather than the solution of a realistic business case. Therefore, the results shown here should not be interpreted as being capable of providing business-critical decision support.

Another important remark is that the standard baseline CC is based on a controller design whose major objective has neither been the reduction of fatigue, nor the maximization of a profit metric. Therefore, benefits of the MPCs with respect to CC in regard of these metrics are to be expected, and should not be considered as key findings of the present work. This CC is utilized here because it has been used for similar comparisons in previous publications (Schlipf et al., 2013; Luna et al., 2020), and therefore allows for some cross-comparisons to these other works. The more sophisticated – and thus relevant – comparison controller formulation is the TTVP MPC, which also has been utilized in other publications (Schlipf et al., 2013; Gros et al., 2013; Luna et al., 2020).

5.2 Tuning

Each element of the controller toolchain (lidar simulator, lidar processing, Moving Horizon Estimator, Model Predictive Controller) comprises a set of tunable parameters, which all impact the control performance. A comprehensive overview of these parameters is provided in Schlipf et al. (2018). Instead of tuning all parameters at once (monolithic approach), the sequential approach of Schlipf et al. (2018) is pursued. Here, the elements are tuned sequentially according to their individual performance criteria.

5.2.1 Tuning of lidar simulator

In Schlipf et al. (2018), the same wind turbine plant model (NREL 5MW onshore) and lidar simulator are used as in the present work. There, the parameters of the lidar simulator are tuned in order to maximize the measurement coherence bandwidth for the rotor-effective wind speed. In other words, the smallest detectable eddy size is maximized, reaching a value of $D_{\mathrm{eddy,min}} = 1.58D = 199$m. Since control of fatigue – and not lidar tuning – is the focus of the present work, the parameters of the lidar simulator in Table 1 are adopted from Schlipf et al. (2018). This "Default lidar" scenario with a low decay constant of 0.1 is accompanied by a "High decay lidar" scenario, where the exponential decay constant is increased to 0.4 (Schlipf, 2016).

5.2.2 Tuning of lidar processing

The raw rotor-effective wind speed from the lidar simulator has to be buffered in order to compensate time delays, and filtered to remove uncorrelated high-frequency information. The buffer and filter parameters have to be tuned. In Schlipf et al. (2018), the tuning of the lidar processing is performed using the reduced wind turbine model as plant model for performance reasons. However, simulations in Loew et al. (2019) have shown that unrealistically high fatigue reduction is possible if the MPC-

Table 1. Parameters of lidar simulator for the "Default lidar" and "High decay lidar" scenarios.

Parameter	Unit	"Default lidar"	"High decay lidar"
Type of lidar	-	Pulsed	Pulsed
Opening angle	[deg]	11.3	11.3
Distance to closest scanning plane $d_{\text{scan,close}}$	[m]	40	40
Distance to farthest scanning plane	[m]	280	280
Number of scanning planes	[-]	10	10
Scanning rate f_{scan}	[Hz]	4	4
Gaussian - Full width at half maximum	[m]	20	20
Gaussian - Evaluation points	[m]	-10,0,10	-10,0,10
Exponential decay constant of turbulent wind	[-]	0.1	0.4

internal and the plant model are matching. Thus, in the present work, the mid-fidelity OpenFAST model is used for tuning of the lidar processing, in order to benefit from its more realistic fatigue behavior.

Buffering: Inspired by Schlipf (2016), the raw rotor-effective wind speed is buffered by an adaptive buffer time span of

$$T_{\text{buffer}}(t) = T_{\text{travel}}(t) - 0.5T_{\text{scan}} - T_{\text{filter}}. \qquad (12)$$

Here, the traveling time $T_{\text{travel}}(t) = d_{\text{travel}}(t)/V_{\text{mean}}(t)$ from the closest scanning plane to the rotor is obtained by the traveling distance d_{travel} and the current mean wind speed V_{mean}. The traveling distance $d_{\text{travel}}(t) = d_{\text{scan,close}} + d_{\text{T}}(t)$ is the nominal distance corrected by the current tower tip deflection. The total scan time $T_{\text{scan}} = 1/f_{\text{scan}}$ is obtained from the scanning rate. The filter delay T_{filter} is zero here, since a zero-phase filter is utilized.

Filtering: Since the lidar correlation varies with wind speed (Schlipf, 2016), the uncorrelated high-frequency information has to be processed by a low-pass filter, which is adaptive as well. In order to avoid the above mentioned filter-delay compensation, only zero-phase algorithms have been considered. Particularly, a zero-phase forward-backward IIR-filter based on a first-order Butterworth filter (function `filtfilt` in `MATLAB`) has been compared to a central moving mean filter (function `movmean` in `MATLAB`). For this purpose, the lidar simulator parameterization from Sect. 5.2.1 and a reasonable initial parameterization of the MHE and MPC have been utilized in turbulent simulations. Despite its simplicity, with different MPC formulations and horizon lengths, the moving mean filter has exhibited superior performance, and thus has been chosen for the present application. It should be noted that this superiority of the moving mean filter has only been observed for the present lidar and wind turbine configuration, and cannot be generalized. Thus, for another configuration, the comparison should be repeated. Beneficially, the moving mean filter only requires tuning of its window length. The empirical formula

$$T_{\text{movmean}}(t) = \frac{D_{\text{eddy,min}}}{V_{\text{mean}}(t)}, \qquad (13)$$

which is based on the smallest detectable eddy size and the current mean wind speed, has led to a very good adaptive tuning for the present setup.

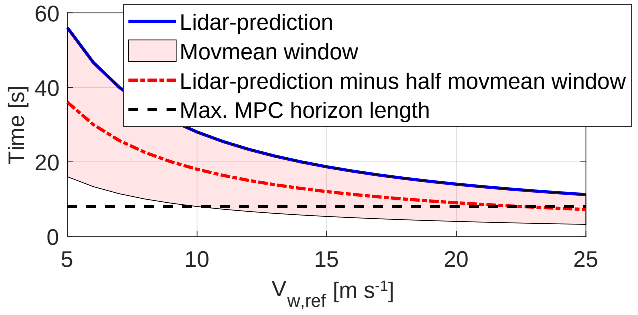


Figure 6. Comparison of the effectively available filtered prediction information (red dash-dotted) to the MPC horizon length (black dashed).

Due to its nature, the central moving mean filter requires sufficient information from the past and future. Except for the beginning of a simulation, the amount of past information is typically sufficient and even growing in the course of the simulation. In contrast, sufficient future information beyond the prediction horizon is only ensured if the inequality

$$T_{\mathrm{pred}}(t) - 0.5T_{\mathrm{movmean}}(t) \geq T_{\mathrm{horiz}} \tag{14}$$

holds, where the lidar-predicted time T_{pred} subtracted by half the moving mean filter length T_{movmean} exceeds the MPC horizon length T_{horiz}. The lidar-predicted time $T_{\mathrm{pred}}(t) = d_{\mathrm{scan,far}}/V_{\mathrm{mean}}(t)$ depends on the distance of the farthest scanning plane to the rotor and the current mean wind speed. As shown in Fig. 6, the inequality typically holds and is only slightly violated at $V_{\mathrm{mean}} > 22\mathrm{ms}^{-1}$.

Further scaling of buffer and filter parameters: In order to verify the above adaptation formula, simulations have been executed where these buffer and filter window lengths are increased or decreased. These results are generated for the "High decay lidar" scenario, where lidar data quality is lower and thus correct filtering is more important. The simulations reveal that the above adaptation laws lead already to high revenue and low fatigue cost. As shown for the filter window length in Fig. 7, for both TTVP and PORFC with scaling factors of 1, a high revenue is maintained while achieving low fatigue cost. Lower scaling factors (shorter filter windows) have a tendency towards higher actuator usage, while higher scaling factors dramatically increase fatigue cost. Consequently, scaling of the buffer and filter parameters is not applied in the present work.

5.2.3 Tuning of the MHE algorithm

The MHE is set up with an estimation horizon length of 8s and a sample time of 0.1s, which are fixed for all MPC configurations.

5.2.4 Tuning of the MHE cost weights

Practical experience has shown that the goal of the MHE should not be an accurate reconstruction of the unmeasured true plant states of tower tip deflection and velocity, as shown in Fig. 8. In fact, an accurate reconstruction would contain high

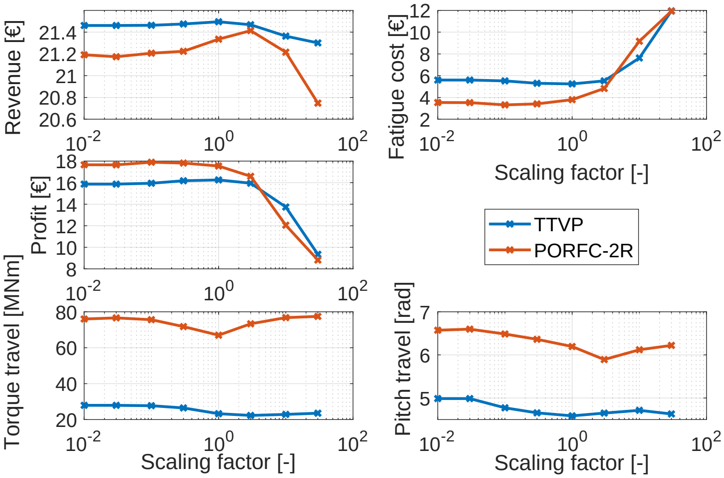


Figure 7. Variation of KPIs depending on the scaling of moving mean filter window length.

frequencies from the plant higher-order degrees of freedom that, not being matched by corresponding degrees of freedom of the MPC-internal model, would spill over and pollute its lower-order degrees of freedom.

395 Instead, the MHE should provide low-frequency initial states for the low-frequency MPC-internal model ("estimated initial" states). Consequently, the MHE is tuned to estimate state trajectories that best fit the behavior of the *reduced model*. This is achieved by setting low weights for the tower variables in the weighting matrix $\boldsymbol{W}_{\mathrm{meas}}$, as shown in Table 2. These low weights allow for significant deviations from the measured tower acceleration and the lidar-based references, and thus for a greater focus on the reduced model dynamics. For tower deflection and velocity, very low values of 10^{-4} are chosen, since
400 these quantities are not measured. For tower acceleration, an intermediate value of 10^{-2} is chosen, since it is measured but its trajectory does not need to be tracked carefully.

The estimated tower deflection in Fig. 8 exhibits a significant steady state offset from the unmeasured true tower deflection. This can be possibly attributed to the inaccurate system model, since this offset also pertains in the MPC prediction, which is based on the same system model.

405 By the intermediate weights in the weighting matrix $\boldsymbol{W}_{\mathrm{prev}}$, the current state trajectories are only loosely tied to the previous ones.

In the weighting matrix $\boldsymbol{W}_{v}$, significant noise is only permitted for the pitch angle and torque rate, in order to enable a close match of pitch and torque estimations with their already accurate measurements.

5.2.5 Tuning of the MPC algorithm

410 The controller sample time is set to 0.1s like in Bottasso et al. (2014) and Gros and Schild (2017). The maximum horizon length of $T_{\mathrm{horiz,max}} = 8$s is chosen based on the findings of Loew et al. (2020b), which indicate that a considerable portion of

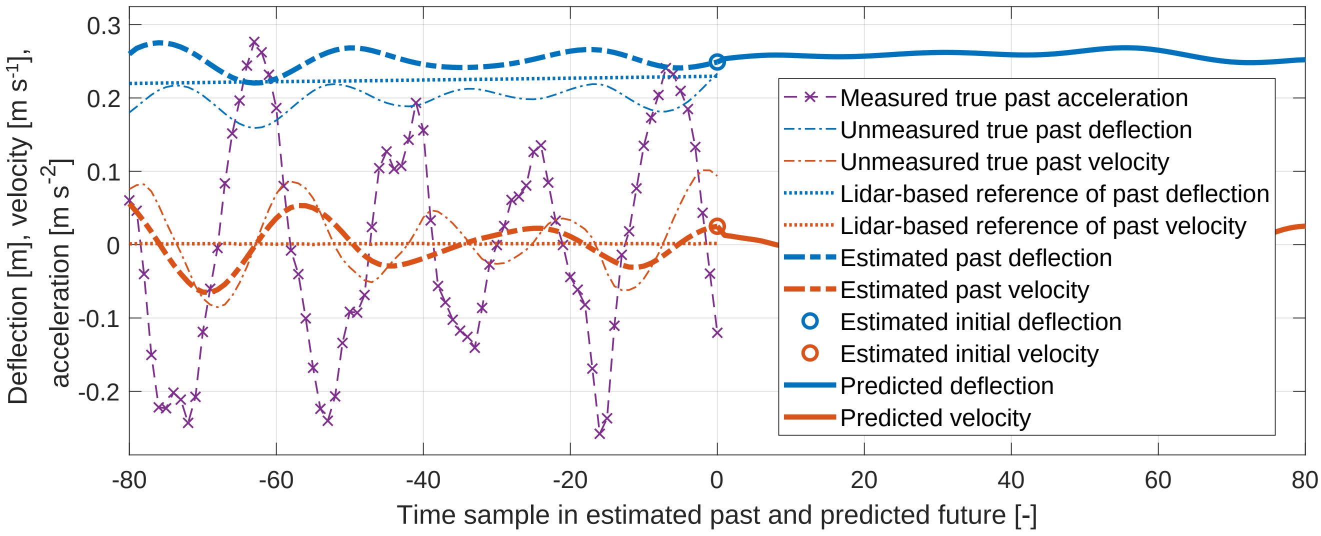


Figure 8. Wind turbine tower quantities ordered by occurrence in the estimation and control process: measured, unmeasured, lidar-based reference (see Sect. 4.2), estimated, and predicted. Negative time samples = estimation horizon of the MHE; positive time samples = prediction horizon of the MPC.

Table 2. Diagonal elements of the weighting matrices $\boldsymbol{W}$ penalizing the corresponding entries of the estimation $\boldsymbol{y}_{\text{est}}$ and noise $\boldsymbol{v}$ vectors.

	$\omega_{\text{r}}(t)$	$d_{\text{T}}(t)$	$\dot{d}_{\text{T}}(t)$	$\beta_{\text{b}}(t)$	$\dot{\beta}_{\text{b}}(t)$	$T_{\text{g}}(t)$	$\ddot{d}_{\text{T}}(t)$
$\boldsymbol{W}_{\text{meas}}$	1	10^{-4}	10^{-4}	1	1	1	10^{-2}
$\boldsymbol{W}_{\text{prev}}$	10^{-2}	10^{-2}	10^{-2}	10^{-2}	10^{-2}	10^{-2}	-
$\boldsymbol{W}_{v}$	1	1	1	1	10^{-3}	10^{-3}	-

the plant stress cycles will be contained in this prediction horizon. However, since horizon length has a substantial impact on performance and – moreover – the longest horizon is not always the best, shorter horizons of $T_{\text{horiz}} = \{4;2;1\}$s will be tested throughout all turbulent studies.

415 One QP is solved per MPC step. The Hessian matrix is automatically convexified to account for possible numerical issues due to the highly non-standard cost formulation of PORFC. Practical experience has shown that performance is improved if the Newton step length of the QP is reduced from 1 to 0.1. This can be explained by the frequently changing optimization problems, especially for PORFC. In this case the initialization of the QP might not be sufficiently close to the optimum, and full Newton steps could leave the region of validity of the quadratic approximation of the Nonlinear Program (Diehl and Gros, 420 2020).

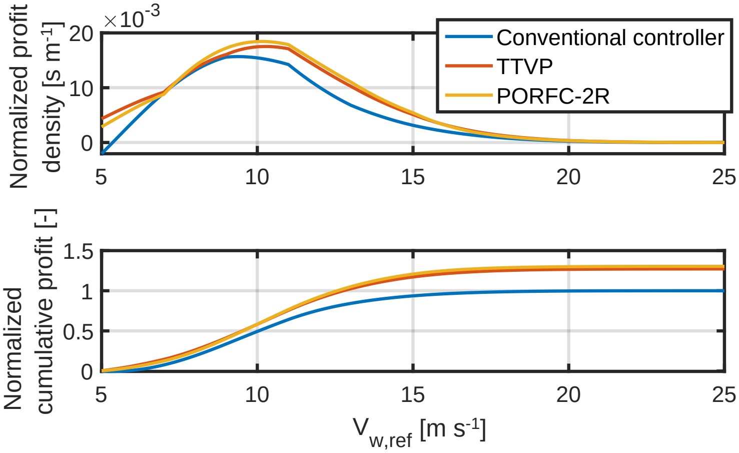


Figure 9. Normalized profit density and cumulative profit. The latter is normalized with respect to the total profit of the conventional controller.

5.2.6 Tuning of the MPC cost weights

As shown in Sect. 3.5, all weights in the cost function except for the fatigue weight $\alpha_{\mathrm{fatigue}}$ are pre-defined for simplicity. Thus, only the fatigue weight has to be tuned in the following. Tuning is executed at a single reference wind speed of $V_{\mathrm{w,ref}} = 9\mathrm{ms}^{-1}$. This wind speed is chosen since, for the conventional controller CC, the highest profit contribution occurs there, as shown in terms of *profit density* in Fig. 9. Profit density represents the incremental contribution to total cumulative profit at a certain wind speed. Other meaningful criteria for a suitable tuning wind speed would be the wind speed where half of total cumulative profit is reached, or where the highest revenue contribution occurs.

The variation of important KPIs with fatigue weights is shown in Fig. 10. For brevity, only the results for the maximum MPC horizon length of 8s are shown. All variants of PORFC are able to maintain high revenue levels while decreasing fatigue cost with increasing fatigue weights. In contrast, the revenue of TTVP declines rapidly above a certain fatigue weight. Thus, the tuning of PORFC can be considered as less critical than that of TTVP. Since in most cases torque travel and pitch travel increase with fatigue weight, low fatigue weights should be preferred as long as fatigue cost is not harmed significantly. Following this strategy, the fatigue weights are determined for all controller formulations and horizon lengths, as shown in Table 3.

Table 3. Optimum fatigue cost weights $\alpha_{\mathrm{fatigue}}$ for different controller formulations and prediction horizon lengths.

Controller formulation	$T_{\mathrm{horiz}} = 8\mathrm{s}$	$T_{\mathrm{horiz}} = 4\mathrm{s}$	$T_{\mathrm{horiz}} = 2\mathrm{s}$	$T_{\mathrm{horiz}} = 1\mathrm{s}$
TTVP	10^4	10^3	10^4	10^5
PORFC-2R	10^4	10^3	10^2	10^3
PORFC-2	10^6	10^3	10^4	10^4
PORFC-5	10^5	10^5	10^5	10^4

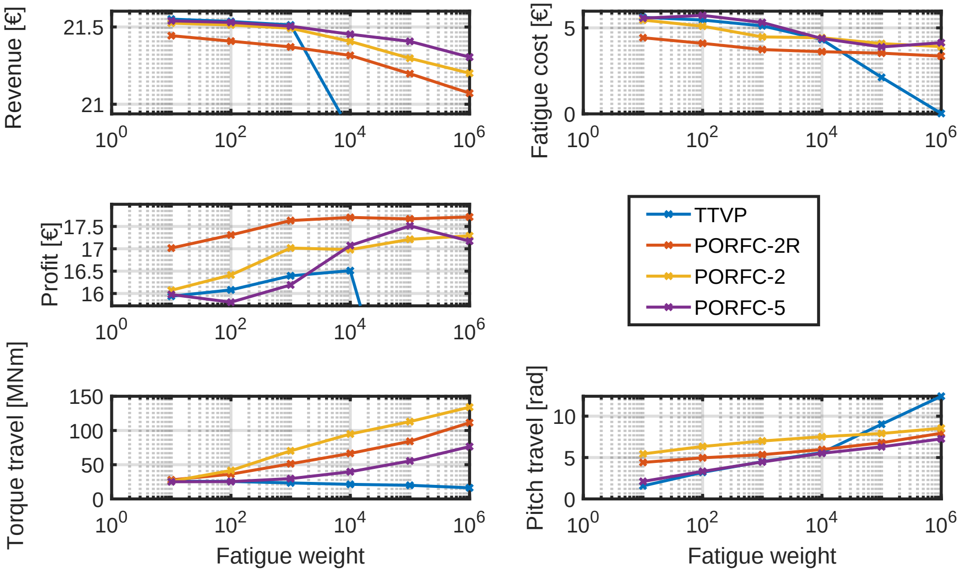


Figure 10. Variation of KPIs for different fatigue weights and a prediction horizon length of $T_{\mathrm{horiz}} = 8\mathrm{s}$.

5.3 Results of turbulent simulations

435 ### 5.3.1 Comparison of controller formulations in the "Default lidar" scenario

As a next step, the optimal tuning weights from Sect. 5.2.6 are fixed and simulations at different reference wind speeds $V_{\mathrm{w,ref}} = \{5; 7; ...; 25\}\mathrm{ms}^{-1}$ are performed for each controller formulation and prediction horizon length. The simulations result in the Weibull-weighted cumulative KPIs shown in Fig. 11.

For all MPCs, revenue is at least slightly below the one of CC. However, especially for PORFC-2R and PORFC-5 with a 440 prediction horizon of $T_{\mathrm{horiz}} \geq 2\mathrm{s}$, the revenue losses remain moderate. In contrast, all MPCs with $T_{\mathrm{horiz}} \geq 2\mathrm{s}$ exhibit substantially lower fatigue cost than CC. As a rough indication of the combined effect, the changes in revenue and fatigue cost can be assessed in terms of the simplified profit indicator. The highest profit gain is achieved by PORFC-2R at maximum prediction horizon, which surpasses CC by 30% and the best TTVP by 2.5%. At least for the present setting, very short horizons of 1s cannot be recommended, since they significantly decrease revenue and greatly increase fatigue.

445 Over different horizon lengths, PORFC-2R exhibits very stable revenue and fatigue levels. In contrast, for the PORFC formulations without residue (PORFC-2, PORFC-5), a shorter horizon of $T_{\mathrm{horiz}} = 4\mathrm{s}$ exhibits higher profit than $T_{\mathrm{horiz}} = 8\mathrm{s}$. This phenomenon may be explained by a higher influence of the prediction errors: due to model errors and wind evolution, the predicted states at the end 4s$< t \leq$8s of a long horizon may be affected by large errors. Since PORFC-2/5 have to rely solely on the predictions, their performance may suffer from long horizons.

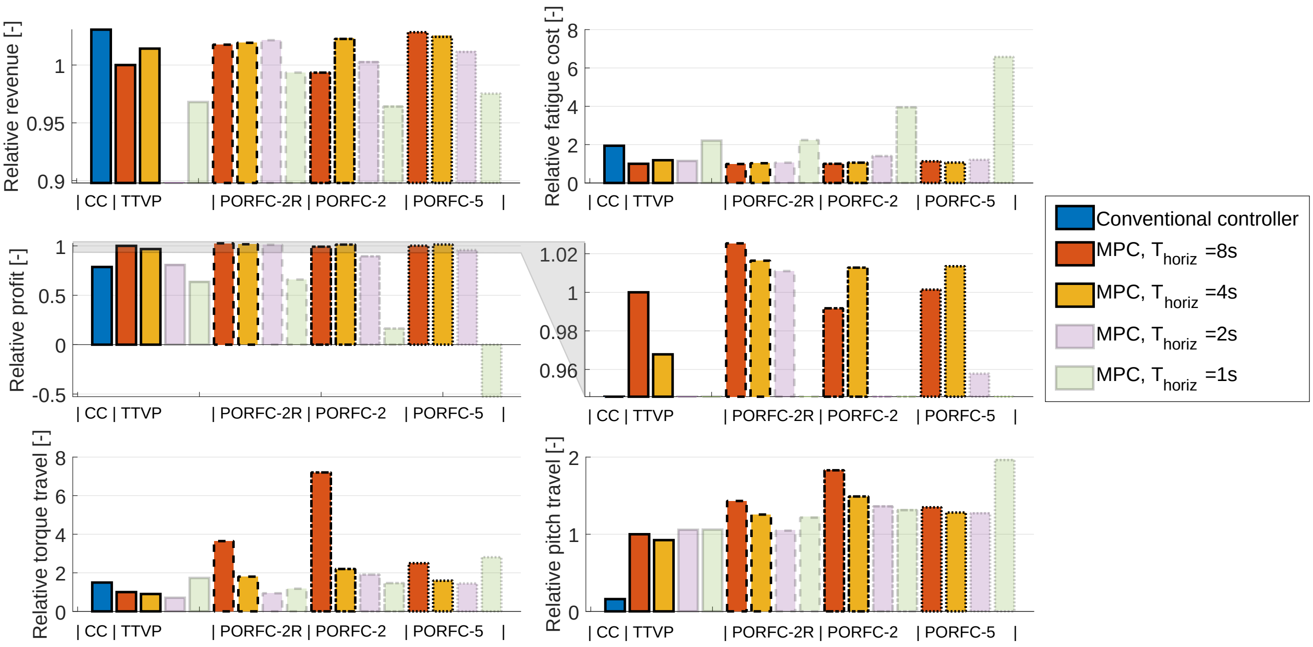


Figure 11. "Default lidar" scenario. Weibull-weighted KPIs for different controller formulations (indicated by edge style) and MPC prediction horizon lengths (indicated by color). The results for the shorter horizons are transparent in order to focus the attention on the more important longer horizons. Results are normalized with respect to the best TTVP configuration with a prediction horizon length of $T_{\mathrm{horiz}} = 8\mathrm{s}$. Middle row, right: Zoomed version of profit plot.

450 For all MPCs, the fatigue reduction with respect to CC comes at the price of a higher pitch travel. Here, TTVP exhibits around 6 times the pitch travel of CC. Since in the literature more moderate increases (e.g., by a factor of 2) are reported (Luna et al., 2020), further studies on pitch penalization should be conducted. For PORFC, pitch travel is even slightly higher, and is somewhat reduced by a reduction of horizon length. Torque travel exhibits a different behavior, where several MPC formulations have a lower torque travel than CC.

455 A look at the profit density and cumulative profit in Fig. 9 shows that both MPC formulations (TTVP, PORFC) "earn money" very similarly over wind speed. Compared to CC, the MPC fatigue reduction strategies lead to profit benefits at very low and at intermediate wind speeds. With the present tuning, TTVP has a slight extra advantage at very low wind speeds, while PORFC-2R is superior over a broad range of intermediate wind speeds.

5.3.2 Performance in the "High decay lidar" scenario

460 The "Default lidar" scenario of the previous sections can be considered as very favorable for lidar-assisted control, since the wind does not change very much between the lidar measurement planes and the rotor. Thus, the lidar provides a fairly good

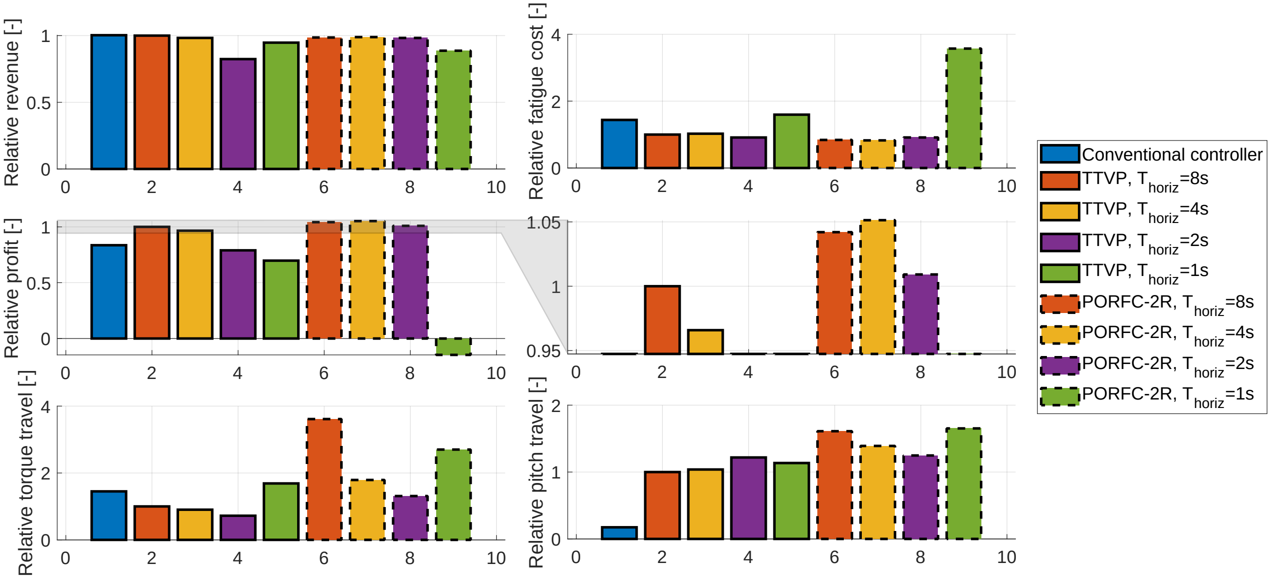


Figure 12. "High decay lidar" scenario. Weibull-weighted KPIs for different controller formulations and prediction horizon lengths. Results are normalized with respect to the best TTVP configuration with a prediction horizon length of $T_{\mathrm{horiz}} = 8\mathrm{s}$. Middle row, right: Zoomed version of profit plot.

estimate of the true incoming wind. In order to challenge the MPCs even more, a further assessment is performed for the "High decay lidar" scenario (see Table 1).

Despite the significant reduction of lidar signal quality, the profit benefit of the MPCs over CC decreases only slightly, as shown in Fig. 12. Particularly, the best-performing PORFC-2R ($T_{\mathrm{horiz}} = 4\mathrm{s}$) still surpasses CC by 26% for the "High decay lidar" scenario, in comparison to the above mentioned 30% for the "Default lidar" scenario.

The profit benefit of 5.1% of the best PORFC-2R ($T_{\mathrm{horiz}} = 4\mathrm{s}$) over the best TTVP ($T_{\mathrm{horiz}} = 8\mathrm{s}$) shows that PORFC-2R is particularly strong in handling situations of low lidar data quality. In a direct comparison using the same horizon length ($T_{\mathrm{horiz}} = 4\mathrm{s}$), PORFC-2R is superior by almost 9%. Just like in the "Default lidar" scenario, PORFC-2R exhibits better revenue stability over the horizon lengths $T_{\mathrm{horiz}} = \{8;4;2\}\mathrm{s}$, but also exhibits excessive fatigue cost at $T_{\mathrm{horiz}} = 1\mathrm{s}$.

5.3.3 Performance in the "Perfect prediction" scenario

Increasing benefit of PORFC-2R with respect to TTVP with lower lidar data quality, conversely suggests decreasing benefits with very high lidar data quality. This hypothesis is actually partially confirmed by the extreme scenario of a perfect wind prediction (without lidar errors). As shown in Fig. 13, as expected, fatigue cost can be reduced by TTVP and PORFC-2R even further than in the "Default lidar" scenario of Sect. 5.3.1. However, relative to TTVP, the advantage of PORFC-2R decreases significantly. For the maximum horizon length, the profit of TTVP even slightly surpasses PORFC-2R by 0.4%. On the other

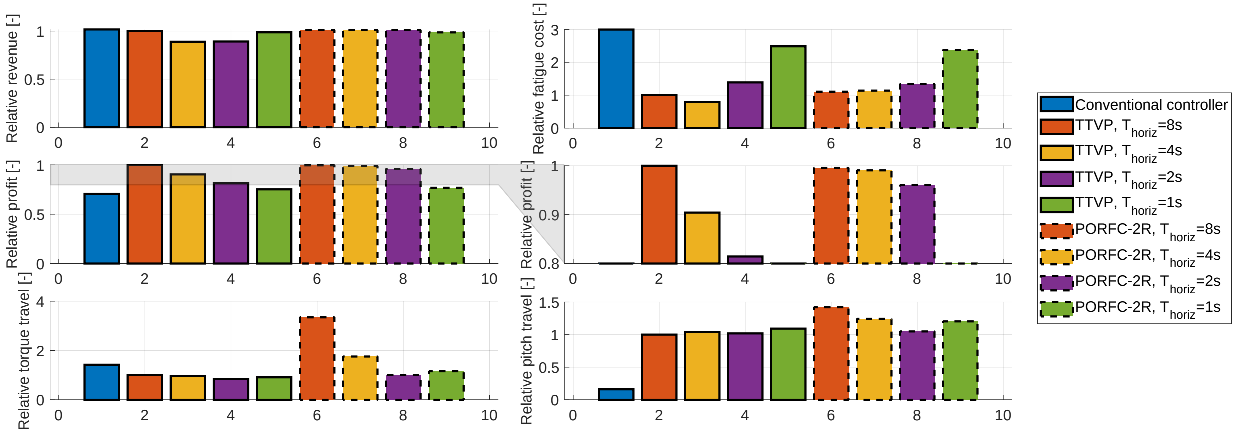


Figure 13. "Perfect prediction" scenario. Weibull-weighted KPIs for different controller formulations and prediction horizon lengths. Results are normalized with respect to the best TTVP configuration with a prediction horizon length of $T_{\mathrm{horiz}} = 8$s. Middle row, right: Zoomed version of profit plot.

hand, as revenue and fatigue cost of PORFC-2R are more stable for shorter horizons, this formulation retains a significant advantage there.

5.3.4 Benefit of "Perfect prediction" vs. "Perfect persistence"

All previous scenarios assumed a wind preview. However, to date, lidar systems can still account for a significant portion of the capital and operational expenditures of wind turbines (Canet et al., 2020). In order to avoid lidar-related costs or effort, some studies are directed towards predictive control without explicit preview measurement (Evans et al., 2015; Jassmann et al., 2016). In this case, the wind prediction over the MPC horizon can for instance be generated via constant extrapolation of the instantaneous wind estimate at the rotor ("persistence"). This motivates an analysis of how the novel PORFC MPC actually benefits from a predictive preview compared to a persistent preview.

Since the design of a rotor-effective wind speed estimator is out of scope of the present work, a "Perfect persistence" scenario is employed and compared to the above "Perfect prediction" scenario.

As shown in Fig. 14, all MPC formulations significantly benefit from prediction (instead of persistence). For all formulations, this is primarily achieved by high fatigue reduction. At the same time, actuator usage is moderately decreased or even increased at some horizon lengths for TTVP, but is significantly decreased for PORFC-2R if $T_{\mathrm{horiz}} \geq 2$s.

These results further indicate the technical benefit of lidar-assisted control, and motivate further studies comparing the realistic lidar wind preview with a sophisticated wind speed estimator.

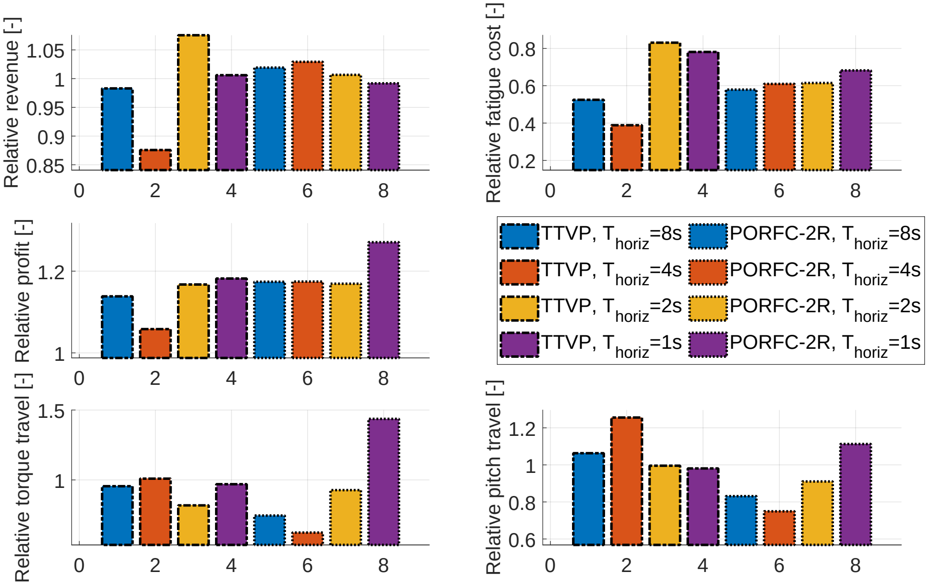


Figure 14. Weibull-weighted KPIs of the "Perfect prediction" scenario normalized with respect to the individually corresponding KPIs of the "Perfect persistence" scenario with same MPC formulation and horizon length.

5.4 Insights into PORFC

In order to gain deeper insight into the behavior of PORFC, short time periods within a turbulent simulation in the "Default lidar" scenario are analyzed.

5.4.1 Evolution of residue

Figure 15 shows a situation where the stress prediction at the initial value $\sigma(t_0)$ turns from a mildly rising slope (MPC step 1) to a mildly falling slope (MPC step 2). Consequently, a new stress maximum is formed and added to the "right-hand-side" of the residue set at MPC step 2, as shown in Fig. 15a. In the following steps 3 to 5, the size of the residue set remains constant; only the right-hand-side value is updated by the current initial stress value.

5.4.2 Evolution of PORFC parameters

As shown in Fig. 15b, the change of extrema in the stress prediction over the course of MPC steps leads to frequent changes in the PORFC stress references (see Sect. 3.4.2) or, more in general, in the PORFC parameter structure (see bars in Fig. 15b). However, since many of the emerging or vanishing stress cycles are small in amplitude, also their corresponding stress reference values are close to the stress prediction trajectory (compare bars to the solid blue line), and thus have low impact on the overall optimization problem.

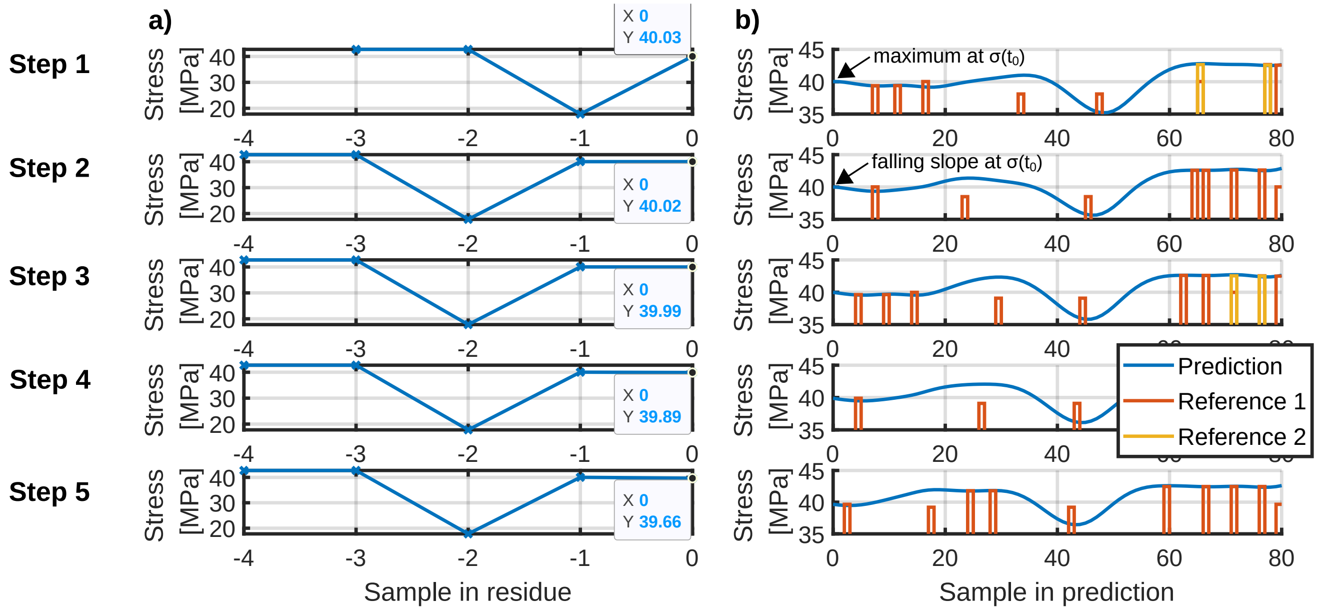


Figure 15. Stress trajectories for PORFC-2R in turbulent wind 10s after the start of the simulation. Top-down: Five consecutive MPC steps. Residue set of variable size, where the values of the last stress samples are labeled (a). Stress prediction of the MPC, and stress references as part of the PORFC parameter set (b).

Since by nature of MPC the stress trajectory is shifted to the left-hand-side with each simulation step, also the PORFC parameter samples are shifted. This becomes even more clear in Fig. 16, where stress reference 1 is plotted over the prediction horizon and over MPC steps (simulation time). Here, over the course of MPC steps, the stress reference pattern is evolving smoothly towards the beginning of the prediction horizon. While some references emerge within the prediction horizon, many references originate at the end and do not vanish before reaching the beginning of the prediction horizon.

5.5 Results of deterministic gust simulations

According to the current standards (IEC, 2005), a central qualification criterion for controllers is their reaction to deterministic gusts. Previous literature already has shown that deterministic gusts are an easy but unrealistic task for predictive controllers like MPCs (Schlipf and Raach, 2016), resulting in too optimistic conclusions regarding extreme load reduction. Besides, extreme loads are not even always design-driving for some wind turbine components (Canet et al., 2020). Nonetheless, the study of gust scenarios sheds additional light on the controller dynamic behavior.

Thus, in the following, the conventional controller is compared to the MPC formulations of TTVP and PORFC-2R in an "Extreme operating gust" scenario (IEC, 2005), with a duration of 10.5s and an initial wind speed that is 2ms^{-1} below rated wind speed ($V_{\text{rated}} = 11.4\text{ms}^{-1}$). In order to test the MPCs with partial knowledge of the gust, a prediction horizon of 4s is chosen. Besides this limited horizon, a perfect wind prediction without lidar errors is assumed.

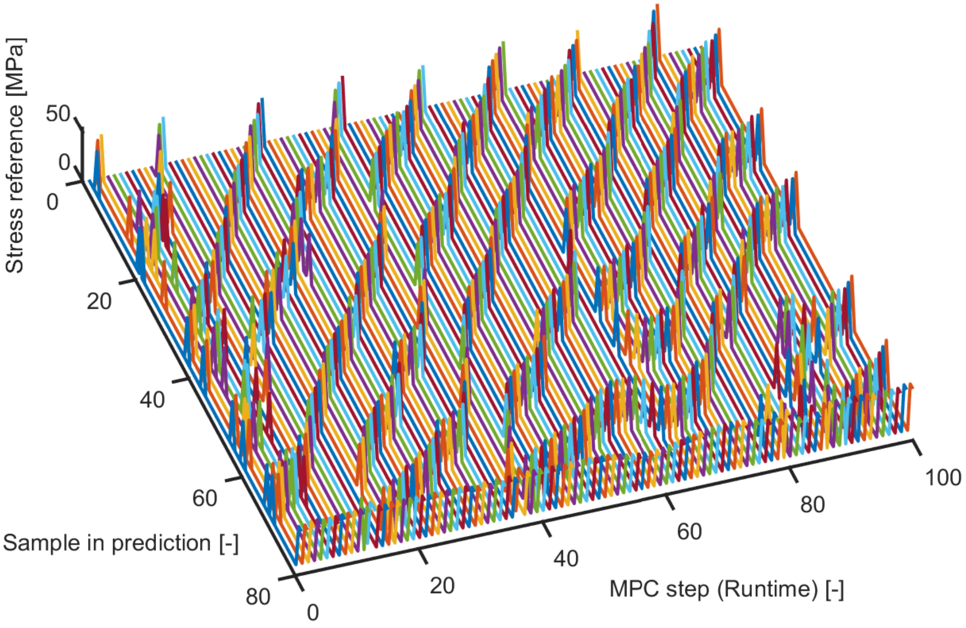


Figure 16. PORFC stress reference over the prediction horizon of 80 samples for 100 consecutive MPC steps.

As shown in Fig. 17, even during the gust, for all controllers the rotor speed remains below the rated speed of 1.267 rad s^{-1}. As a result, the conventional controller remains at the minimum pitch angle, and the tower deflection freely follows the gust wind speed, which leads to a high positive excursion. After the gust, the tower oscillation quickly vanishes due to aerodynamic damping. 525

In contrast, the MPCs anticipate the incoming gust, and react to a significant extent by pitching the blades. Interestingly, despite their different fatigue cost formulations, the MPCs exhibit very similar pitching behavior. As expected, the TTVP and the PORFC-2R MPC attenuate very effectively the tower excursion, and dampen the oscillation immediately. However, PORFC-2R puts less priority on the attenuation of the tower excursion. Since the tower deflection has been flat prior to the gust, the stress residue of PORFC-2R contains only stress values around the steady state. Consequently, PORFC-2R assumes only a small stress cycle with low damage potential during the gust. This behavior is changed, if the stress residue is initialized with 0 MPa, which corresponds to an undeflected tower prior to operation. Due to this stress memory, PORFC-2R identifies a large half cycle, and consequently tries to further limit the maximum tower excursion by peak-shaving, as shown by the purple trajectory in Fig. 17.

535 Clear differences of PORFC-2R with respect to TTVP can be seen in the rotor speed and generator power dynamics. At the beginning of the gust, the generator power is reduced in order to achieve a high rotor speed during the gust. This behavior can be attributed to an attempt at harvesting the energy of the gust, and also has been observed for an MPC where 5 QPs

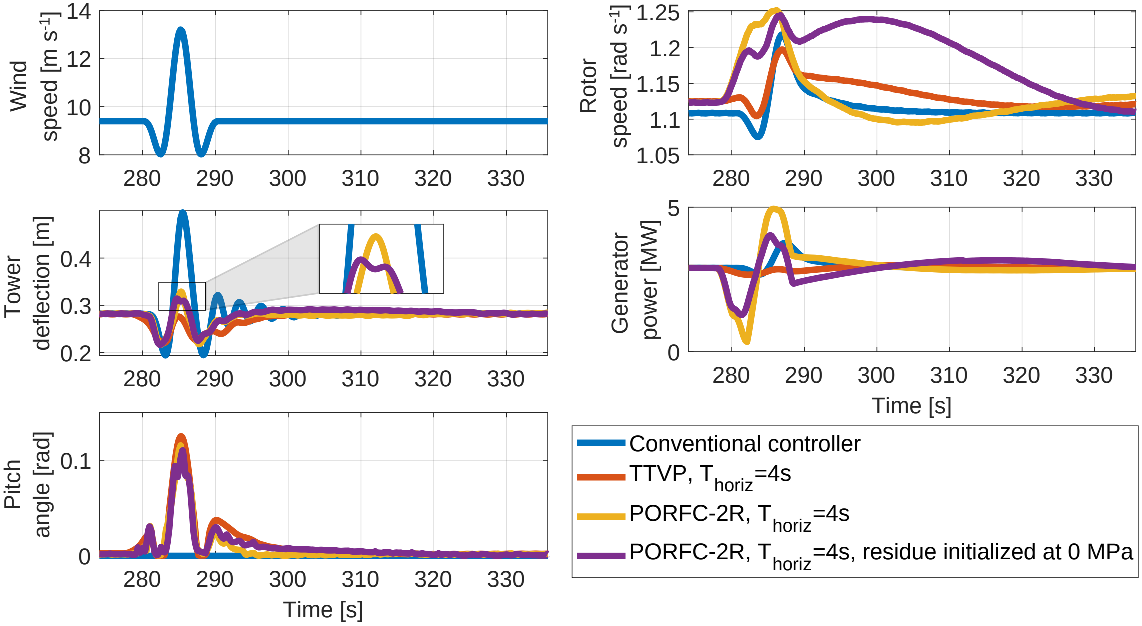


Figure 17. Extreme operating gust at $V = V_{\text{rated}} - 2\text{ms}^{-1}$.

(instead of 1 QP) have been solved per MPC step for better convergence. For the PORFC-2R MPC with the 0 MPa residue, the rotor speed remains at a high level for a longer time frame, which is an unusual behavior and requires more investigation. 540 Finally, it can be noted that the steady state rotor speed is slightly higher for the MPCs than for the conventional controller, as seen before the gust. Assuming perfect tracking of optimal rotor speed by the conventional controller, this difference can be attributed to the MPC plant-model mismatch. However, as shown in Fig. 17, the rotor speed difference does not result in significant suboptimality of the steady state power capture.

6 Conclusions and outlook

545 6.1 Conclusions

The present work represents a significant step in assessing the benefits of the MPC formulation of PORFC. For this purpose, the simulation setup of Loew et al. (2020b) has been extended by a realistic lidar simulator, lidar processing, and an MHE.

First, the PORFC formulation has been presented in an application-focused way. It has been highlighted how PORFC directly incorporates mechanical fatigue in predictive wind turbine control. Since fatigue requires long observation windows, stress 550 history has been considered in a consistent manner by carrying along a *residue* (PORFC-2R).

Second, the formulation of the MHE has been explained, where the lidar wind estimate has been used to generate an initialization for the unmeasured tower states.

Third, a sequential tuning approach has been employed for the lidar simulator, lidar processing, MHE, and MPC:

- For the lidar simulator, parameters from the literature have been utilized, which maximize the measurement coherence bandwidth.
- For the lidar buffering and filtering, simple adaptive tuning laws have been employed. Simulations have revealed that they result already in good performance, and that no further tuning seems to be required.
- For the MHE, instead of accurately reconstructing the plant states, the cost weights have been tuned to estimate only the low-frequency state information that can be handled by the MPC-internal model.
- For the MPC, four different prediction horizon lengths have been employed throughout the study, since no single horizon length has led to the best performance in all scenarios. In the MPC cost function, the fatigue weight has been tuned systematically for each controller formulation and horizon length.

Finally, extensive economic and dynamic simulation results have been presented for turbulent and gust wind settings:

- In the "Default lidar" scenario, all MPCs are able to significantly reduce fatigue cost with respect to a conventional PID controller, while PORFC-2R has to sacrifice less revenue than a conventional MPC. For shorter horizons, especially the PORFC formulation with residue has shown a more robust performance than the conventional MPC.
- In the "High decay lidar" scenario with a lower lidar prediction quality, the advantage of PORFC-2R over the conventional MPC even increases. This suggests that PORFC-2R is a recommended solution especially for lower lidar prediction quality.
- In the "Perfect prediction" scenario, both MPCs have exhibited similar results. A comparison to a wind persistence setting has shown that PORFC-2R benefits more from the availability of this high quality prediction than the conventional MPC.
- In all considered turbulent scenarios, MPCs with a very short prediction horizon of 1s have obtained only modest results.
- An excerpt of a turbulent simulation with PORFC-2R has demonstrated how the residue is updated, and that the parametric stress references evolve smoothly following an expected pattern.
- In an Extreme Operating Gust setting, both MPCs have shown a similar pitching behavior and the effective attenuation of tower excursion. During the gust, PORFC-2R has shown a higher variability in the rotor speed.

6.2 Outlook

The MPC formulation of PORFC still has several aspects worth investigating:

- For the MHE tuning, an automated but still computationally tractable approach should be developed.
- The MHE- and MPC-internal system model has a significant error with respect to the plant system. Thus, online model adaptation promises further benefits.
- In the MPC cost function, economic terms for the actuator, blade and drive-train damage should be included.

- In certain business cases, the goal may not be to minimize fatigue but simply to keep the fatigue rate on average below certain thresholds, or to keep the cumulative damage below a threshold by the end of service. To assist these goals, the PORFC cost function could be modified and fatigue could be added as a parametric constraint in the MPC. Alternatively, an outer control loop based on structural health monitoring could be added (Do and Söffker, 2020), which adapts the MPC cost function weights. More in general, other scenario-based more sophisticated and comprehensive cost functions should be considered, to better capture the complex interactions of fatigue damage with the economic utilization of wind assets.
- Instead of the NREL baseline controller (Jonkman et al., 2009), the MPCs could be compared to a more modern reference controller, as for example the one of Abbas et al. (2022).
- The novel PORFC MPC has been extensively simulated and is ready for application on real systems. Consequently, just like for conventional MPCs in Sinner et al. (2021) and Dickler et al. (2021), the novel PORFC MPC should be assessed on scaled and full-scale wind turbines.

Nomenclature

Quantity	Unit	Explanation
$\bar{a}$	-	Quantity sampled on the control intervals of the prediction horizon
$\hat{a}$	-	Quantity fixed for one MPC step
$\tilde{a}$	-	Quantity estimated from measurements

Abbreviation		Explanation
CC	-	Conventional PID controller
lidar	-	Light Detection And Ranging
MHE	-	Moving Horizon Estimator
MPC	-	Model Predictive Controller
PORFC	-	Parametric Online Rainflow Counting
QP	-	Quadratic Programming
RFC	-	Rainflow counting algorithm
TTVP	-	Tower Tip Velocity Penalization

Code and data availability. MATLAB figure files for the lossless extraction of the shown results can be retrieved via the DOI 10.5281/zenodo.6600689. The MATLAB function and a test script for the PORFC parameter generation and the residue update can be retrieved via the DOI 10.5281/zenodo.6600833. Further data can be provided upon request.

Author contributions. SL developed the controller formulation in collaboration with CLB, implemented all elements of the toolchain, and performed the simulation studies. SL and CLB collaborated on the interpretation and analysis of the results. CLB supervised the work. Both authors provided important input to this research work through discussions, feedback and by writing the paper.

Competing interests. CLB is the Editor in Chief of Wind Energy Science. The authors have no other conflict of interests to declare.

Acknowledgements. The authors would like to thank the company sowento GmbH for kindly providing a free license of their lidar simulator. Furthermore, the close and active support of Martin Koch, David Schlipf and Steffen Raach of sowento GmbH is gratefully acknowledged.

References

Abbas, N. J., Zalkind, D. S., Pao, L., and Wright, A.: A reference open-source controller for fixed and floating offshore wind turbines, Wind Energy Science, 7, 53–73, https://doi.org/10.5194/wes-7-53-2022, 2022.

Anand, A.: Optimal Control of Battery Energy Storage System for Grid Integration of Wind Turbines, Master's thesis, TU Munich, Munich, 2020.

ASTM International: Standard practices for cycle counting in fatigue analysis (ASTM 1049-85), https://doi.org/10.1520/E1049-85R17, 1985.

Barradas-Berglind, J. d. J., Wisniewski, R., and Soltani, M.: Fatigue damage estimation and data-based control for wind turbines, IET Control Theory & Applications, 9, 1042–1050, https://doi.org/10.1049/iet-cta.2014.0730, 2015.

Barradas-Berglind, J. J. and Wisniewski, R.: Representation of fatigue for wind turbine control, Wind Energy, 19, 2189–2203, https://doi.org/10.1002/we.1975, 2016.

Bottasso, C. and Croce, A.: Cascading Kalman observers of structural flexible and wind states for wind turbine control: Scientific Report DIA-SR 09-02, Dipartimento di Ingegneria Aerospaziale, Politecnico di Milano, 2009.

Bottasso, C. L., Pizzinelli, P., Riboldi, C., and Tasca, L.: LiDAR-enabled model predictive control of wind turbines with real-time capabilities, Renewable Energy, 71, 442–452, https://doi.org/10.1016/j.renene.2014.05.041, 2014.

Canet, H., Löw, S., and Bottasso, C. L.: Lidar-assisted control in wind turbine design: Where are the potential benefits?, Journal of Physics: Conference Series, 1618, 042 020, https://doi.org/10.1088/1742-6596/1618/4/042020, 2020.

Dickler, S., Wintermeyer-Kallen, T., Zierath, J., Bockhahn, R., Machost, D., Konrad, T., and Abel, D.: Full-scale field test of a model predictive control system for a 3 MW wind turbine, Forschung im Ingenieurwesen, 85, 313–323, https://doi.org/10.1007/s10010-021-00467-w, 2021.

Diehl, M. and Gros, S.: Numerical Optimal Control: Lecture manuscript, University of Freiburg, Freiburg and Gothenburg, https://www.syscop.de/files/2020ss/NOC/book-NOCSE.pdf, 2020.

Do, H. and Söffker, D.: Wind Turbine Lifetime Control Using Structural Health Monitoring and Prognosis, IFAC-PapersOnLine, 53, 12 669–12 674, https://doi.org/10.1016/j.ifacol.2020.12.1847, 2020.

Evans, M. A., Cannon, M., and Kouvaritakis, B.: Robust MPC Tower Damping for Variable Speed Wind Turbines, IEEE Transactions on Control Systems Technology, 23, 290–296, https://doi.org/10.1109/TCST.2014.2310513, 2015.

Gros, S.: An Economic NMPC Formulation for Wind Turbine Control, in: 52nd IEEE Conference on Decision and Control, pp. 1001–1006, IEEE, Firenze, Italy, https://doi.org/10.1109/CDC.2013.6760013, 2013.

Gros, S. and Schild, A.: Real-time economic nonlinear model predictive control for wind turbine control, International Journal of Control, 90, 2799–2812, https://doi.org/10.1080/00207179.2016.1266514, 2017.

Gros, S., Vukov, M., and Diehl, M.: A real-time MHE and NMPC scheme for wind turbine control, in: 2013 IEEE 52nd Annual Conference on Decision and Control (CDC), pp. 1007–1012, IEEE, Piscataway, NJ, https://doi.org/10.1109/CDC.2013.6760014, 2013.

Grüne, L. and Pannek, J.: Nonlinear Model Predictive Control: Theory and Algorithms, Communications and Control Engineering, Springer International Publishing, Cham, Switzerland, 2nd ed. 2017 edn., https://doi.org/10.1007/978-3-319-46024-6, 2017.

Haibach, E.: Betriebsfestigkeit: Verfahren und Daten zur Bauteilberechnung, VDI-Buch, Springer, Berlin, 3., korrigierte und erg. aufl. edn., https://doi.org/10.1007/3-540-29364-7, 2006.

Hau, E.: Windkraftanlagen: Grundlagen. Technik. Einsatz. Wirtschaftlichkeit, Springer Berlin Heidelberg, Berlin, Heidelberg, 6th ed. edn., https://doi.org/10.1007/978-3-662-53154-9, 2017.

Heinrich, C., Khalil, M., Martynov, K., and WEVER, U.: Online remaining lifetime estimation for structures, Mechanical Systems and Signal Processing, 119, 312–327, https://doi.org/10.1016/j.ymssp.2018.09.028, 2019.

Huang, R., Biegler, L. T., and Patwardhan, S. C.: Fast Offset-Free Nonlinear Model Predictive Control Based on Moving Horizon Estimation, Industrial & Engineering Chemistry Research, 49, 7882–7890, https://doi.org/10.1021/ie901945y, 2010.

IEC: IEC 61400-1 Ed.3: Wind turbines - Part 1: Design requirements, 2005.

Jassmann, U., Zierath, J., Dickler, S., and Abel, D.: Model Predictive Wind Turbine Control for Load Alleviation and Power Leveling in Extreme Operation conditions, in: 2016 IEEE Conference on Control Applications (CCA), IEEE, https://doi.org/10.1109/CCA.2016.7587997, 2016.

Jonkman, J., Butterfield, S., Musial, W., and Scott, G.: Definition of a 5-MW Reference Wind Turbine for Offshore System Development: Technical Report NREL/TP-500-38060, National Renewable Energy Laboratory, https://doi.org/10.2172/947422, 2009.

Köhler, M., Jenne, S., Pötter, K., and Zenner, H.: Zählverfahren und Lastannahme in der Betriebsfestigkeit, Springer, Berlin, Heidelberg, https://doi.org/10.1007/978-3-642-13164-6, 2012.

Loew, S. and Obradovic, D.: Formulation of Fatigue Dynamics as Hybrid Dynamical System for Model Predictive Control, IFAC-PapersOnLine, 53, 6616–6623, https://doi.org/10.1016/j.ifacol.2020.12.080, 2020.

Loew, S., Obradovic, D., and Bottasso, C. L.: Direct Online Rainflow-counting and Indirect Fatigue Penalization Methods for Model Predictive Control, in: 2019 18th European Control Conference (ECC), pp. 3371–3376, IEEE, https://doi.org/10.23919/ECC.2019.8795911, 2019.

Loew, S., Obradovic, D., Anand, A., and Szabo, A.: Stage Cost Formulations of Online Rainflow-counting for Model Predictive Control of Fatigue, in: 2020 European Control Conference (ECC), pp. 475–482, https://doi.org/10.23919/ECC51009.2020.9143939, 2020a.

Loew, S., Obradovic, D., and Bottasso, C. L.: Model predictive control of wind turbine fatigue via online rainflow-counting on stress history and prediction, Journal of Physics: Conference Series, 1618, 22 041, https://doi.org/10.1088/1742-6596/1618/2/022041, 2020b.

Löw, S. and Obradovic, D.: Real-time Implementation of Nonlinear Model Predictive Control for Mechatronic Systems Using a Hybrid Model, atp edition, 60, 46–53, https://doi.org/10.17560/atp.v60i09.2359, 2018.

Luna, J., Falkenberg, O., Gros, S., and Schild, A.: Wind turbine fatigue reduction based on economic-tracking NMPC with direct ANN fatigue estimation, Renewable Energy, 147, 1632–1641, https://doi.org/10.1016/j.renene.2019.09.092, 2020.

Manwell, J. F., McGowan, J. G., and Rogers, A. L.: Wind energy explained: Theory, design and application, John Wiley & Sons, Chichester, England, https://doi.org/10.1002/0470846127, 2002.

Miner, M. A.: Culmulative Damage in Fatigue, Journal of Applied Mechanics, 12, A159–A164, https://doi.org/10.1115/1.4009458, 1945.

Raach, S. and Schlipf, D.: Cross-tool realistic lidar simulations: Presentation at the IEA Wind Task 32 workshop on Certification of Lidar-assisted control applications, https://www.ieawindtask32.org/ws08-internal-documents/, 2018.

Rawlings, J. B., Mayne, D. Q., and Diehl, M. M.: Model predictive control: Theory, computation, and design, Nob Hill Publishing, Madison, Wisconsin, 2nd edition edn., 2017.

Ritter, B.: Nonlinear State Estimation and Noise Adaptive Kalman Filter Design for Wind Turbines, Dissertation, Technische Universität Darmstadt, Darmstadt, https://doi.org/10.25534/tuprints-00011785, 2020.

Sanchez, H., Escobet, T., Puig, V., and Odgaard, P. F.: Health-aware Model Predictive Control of Wind Turbines using Fatigue Prognosis, IFAC-PapersOnLine, 48, 1363–1368, https://doi.org/10.1016/j.ifacol.2015.09.715, 2015.

Schlipf, D.: Lidar-assisted control concepts for wind turbines, Dissertation, University of Stuttgart, Stuttgart, Germany, https://doi.org/10.18419/OPUS-8796, 2016.

Schlipf, D. and Raach, S.: Turbulent Extreme Event Simulations for Lidar-Assisted Wind Turbine Control, Journal of Physics: Conference Series, 753, 052 011, https://doi.org/10.1088/1742-6596/753/5/052011, 2016.

Schlipf, D., Schlipf, D. J., and Kühn, M.: Nonlinear model predictive control of wind turbines using LIDAR, Wind Energy, 16, 1107–1129, https://doi.org/10.1002/we.1533, 2013.

Schlipf, D., Fürst, H., Raach, S., and Haizmann, F.: Systems Engineering for Lidar-Assisted Control: A Sequential Approach, Journal of Physics: Conference Series, 1102, 012 014, https://doi.org/10.1088/1742-6596/1102/1/012014, 2018.

Shi, Y., Xu, B., Tan, Y., and Zhang, B.: A Convex Cycle-based Degradation Model for Battery Energy Storage Planning and Operation, in: 2018 Annual American Control Conference (ACC), pp. 4590–4596, IEEE, https://doi.org/10.23919/ACC.2018.8431814, 2018.

Simon, D.: Kalman filtering with state constraints: a survey of linear and nonlinear algorithms, IET Control Theory & Applications, 4, 1303–1318, https://doi.org/10.1049/iet-cta.2009.0032, 2010.

Sinner, M., Petrovic, V., Langidis, A., Neuhaus, L., Holling, M., Kuhn, M., and Pao, L. Y.: Experimental Testing of a Preview-Enabled Model Predictive Controller for Blade Pitch Control of Wind Turbines, IEEE Transactions on Control Systems Technology, pp. 1–15, https://doi.org/10.1109/TCST.2021.3070342, 2021.

The MathWorks Inc.: Rainflow counts for fatigue analysis, https://de.mathworks.com/help/signal/ref/rainflow.html, 2018.

Verschueren, R., Frison, G., Kouzoupis, D., Frey, J., van Duijkeren, N., Zanelli, A., Novoselnik, B., Albin, T., Quirynen, R., and Diehl, M.: acados—a modular open-source framework for fast embedded optimal control, Mathematical Programming Computation, https://doi.org/10.1007/s12532-021-00208-8, 2021.