



Adaptive economic wind turbine control

Abhinav Anand and Carlo L. Bottasso

Technical University of Munich, Wind Energy Institute, Boltzmannstraße 15, 85748 Garching b. München, Germany

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

Abstract.

Model predictive control (MPC) for wind turbines offers several interesting advantages over simpler techniques, as for example the direct optimization of a goal function, the inclusion of constraints, non-linear coupled dynamics, and wind preview (when available). To enable real-time execution, MPC uses a reduced order model (ROM) that approximates the dynamics of the controlled system using only a limited number of degrees of freedom. As a result, the accuracy of the ROM is often the main limit to the performance of MPC. To address this problem, an adaptive controller-internal model can reduce plant-model mismatches, potentially leading to improved performance.

This work proposes an adaptive economic nonlinear MPC (ENMPC) for wind turbines. The controller maximizes profit by optimally balancing fatigue damage cost with revenue due to power generation. The cyclic fatigue cost is formulated directly within the controller using the novel parametric online rainflow counting (PORFC) approach. PORFC provides a rigorous continuous expression of the discontinuous cyclic fatigue cost using time-varying parameters. Adaptivity is obtained by a controller-internal grey-box model that combines reduced order physical dynamics with data-driven correction terms. These are implemented via a neural network that is trained offline. Additionally, system state and disturbance estimators are included in the closed-loop controller.

The improvement in state predictions due to model adaptation is first assessed and compared with respect to the non-adapted baseline ROM in open loop. The performance of the adaptive ENMPC and the impact of a reduced plant-model mismatch is then assessed in closed loop for a reference multi-MW onshore wind turbine in a realistic simulation environment. Results show that the adaptive ENMPC yields higher economic profits at significantly lower pitch and torque travels, compared to the baseline non-adaptive ENMPC. While the enhanced closed-loop performance and economic gains of the proposed model adaptation are significant, they come at the cost of a slight increase in the computational burden of the controller.

1 Introduction

Wind turbine operation and control have recently shifted from the traditional goal of power maximization to more economically driven goals. In this new paradigm, turbines are operated with damage awareness in mind (Barradas-Berglind and Wisniewski, 2016; Gros and Schild, 2017; Loew et al., 2019; Anand et al., 2022; Loew et al., 2023). This shift is driven by the impact of fatigue damage, which shortens the operational life of turbines and increases operation and maintenance (O&M) costs. In fact, these factors are critical to the economic profitability of operators of wind energy assets (Canet et al., 2021; Stehly et al., 2020).



One widely used method for developing economic controllers for wind turbines is model predictive control (MPC) (Rawlings et al., 2017). MPC optimizes control actions over the short-term future by predicting system behavior, and then solving an economic optimization problem based on these predictions (Rawlings et al., 2012). When considering wind energy systems, an economic optimization should balance the conflicting objectives of maximizing revenue from power generation and of minimizing the cost of fatigue-related damage. Additionally, a controller should always guarantee that the system operates within feasible limits. The effectiveness of the controller ultimately depends on the quality of the solution of this constrained optimization, which essentially relies – not only on the economic model – but also on the quality of predictions of the controller-internal model.

The existing literature on the economic control of wind turbines using MPC is based on two main approaches for estimating cyclic fatigue: indirect and direct methods. The indirect approach uses a proxy for fatigue (Barradas-Berglind and Wisniewski, 2016; Gros and Schild, 2017). This method is convenient because it avoids the use of cycle counting that, because of its branching nature, introduces discontinuities in the calculations (Loew et al., 2021). As a result, with the indirect approach gradient-based methods can be used to solve the optimization problem. However, this also means that fatigue is only approximately taken into account through a proxy quantity, which might not always provide for accurate results. The direct approach, on the other hand, estimates fatigue explicitly using online cycle counting directly within the MPC framework (Loew et al., 2020b; Anand et al., 2022). This method, termed parametric online rainflow counting (PORFC), is relatively new and was first introduced in our previous work Loew et al. (2020a). It has since been applied in its fundamental form in various contexts, including wind turbine fatigue control (Loew et al., 2020b), battery cyclic aging control (Loew et al., 2021), and control of grid-connected wind-battery hybrid systems (Anand et al., 2021, 2022). To the authors' knowledge, PORFC is the only approach that allows for the rigorous treatment of fatigue, as it was design to produce the same results on receding horizons that would be obtained by cycle counting the whole response time history a posteriori (Loew et al., 2020a).

In both approaches mentioned above, the controller relies on an internal model of the wind turbine to predict its behavior over the MPC horizon. However, due to computational constraints, the internal representation of the system response is typically obtained through a reduced order model (ROM). ROMs approximate the system dynamics using a limited number of degrees of freedom (DOFs), enabling the real-time execution of the controller. The accuracy of these predictions plays a crucial role in the performance of the controller. A closer match between the predicted and actual wind turbine dynamics allows for the controller to make more precise decisions when optimizing the objective function, while at the same time ensuring compliance with system constraints. In fact, the degree of model mismatch not only affects optimality, but may also impacts the ability of the closed-loop system to operate strictly within admissible or desired limits.

One way to address model mismatches is by using state observers and estimators (Anand et al., 2022; Loew and Bottasso, 2022), where measurements of the response of the wind turbine from previous time steps are used to estimate and adjust the initialization of the controller for the next step. However, the effectiveness of this approach is limited when estimates are obtained from underlying models of scarce accuracy.



60 A more compelling alternative is to adapt the internal model dynamics of the controller, either by using measured operational data or synthetic results from models of sufficiently high fidelity that can accurately represent the plant behavior. One of the aims of this paper is to develop practical methods for enabling adaptivity in wind turbine ROMs.

Data-driven adaptation can be implemented online, offline, or in a hybrid manner, each of these options offering distinct trade-offs in terms of advantages, challenges, and practical applicability to economic control. Online adaptation allows for quickly adjusting model behavior in real time based on observed plant response. However, for industrial applications, it is difficult to imagine how one could guarantee the correct learning of model corrections, and how such a system could be demonstrated to be always safe and certifiable in practice. In contrast, offline adaptation allows for the systematic integration of data, enabling a rigorous verification and validation of any learned correction terms prior to their deployment in the field. A rigorous offline verification and validation opens the door to the certification of adaptive controllers, which must meet strict standards in terms of performance, safety, and reliability before deployment in real-world operations.

In the context of economic MPC, offline model adaptation can be achieved in several ways: by adjusting the parameters of the ROM (Schreiber et al., 2020), or by incorporating a correction term into the ROM or MPC optimization function (Bottasso et al., 2006; Collet et al., 2021; Soloperto et al., 2022), or by a combination of both methods. Since a ROM captures only some aspects of the behavior of the plant, the sole tuning of its parameters is often insufficient and might actually lead to nonphysical values of the parameters. Instead, data can be leveraged to learn and incorporate missing physics into the internal model of the controller. This is indeed the approach followed here.

The key contribution of this work is the development of a novel economic nonlinear model predictive controller (ENMPC) with offline-adaptive capabilities enabled by data-learned ROM corrections. The controller maximizes profit by balancing two competing factors: revenue from power generation and costs associated with fatigue damage of the turbine components. The cost of cyclic damage is formulated using the novel PORFC approach, which performs online rainflow analysis over the stress history and its future predictions to generate time-varying parameters. These parameters are then used to obtain a continuous expression of the discontinuous fatigue cost, which is incorporated into the MPC optimization. The controller-internal model is a grey-box that combines reduced order physical dynamics with offline-computed correction terms. These terms are formulated as neural networks trained on high-resolution measured or synthetic data of the wind turbine. Their objective is to reduce model mismatch, thereby improving online closed-loop performance. The profit formulation used here does not explicitly incorporate the effects of fatigue on component reliability and O&M costs, simply because of a lack of data and appropriate models linking loads with failure rates, instead representing fatigue-related costs solely through the amortization of the overall component cost, as discussed in TotalControl (2022).

The paper is structured as follows. Section 2 introduces the proposed model adaptation approach. It describes both the simplified wind turbine model used as internal ROM, and the data-driven modeling of the correction term. Section 3 provides a detailed formulation of the ENMPC, discussing various aspects of the underlying optimization problem. This section also describes the state and wind speed estimators, which provide initial conditions and disturbance predictions for the ENMPC. Section 4 presents and discusses the results from a case study. The analysis begins with an open-loop assessment of the model adaptation approach, followed by a closed-loop evaluation of its impact on economic performance, actuator usage, and



95 computational burden. Additionally, this section examines the benefits of the proposed model adaptation under different wind input and preview scenarios. Section 5 summarizes the key findings of this work and outlines potential directions for future research.

2 Model adaptation

The proposed model adaptation approach enhances the ROM dynamics by introducing a data-driven correction function, 100 $\Delta \mathbf{F}_{\text{ROM}}(\cdot)$. This function is designed to compensate for the mismatch between the ROM and the actual wind turbine behavior. The adapted model dynamics, denoted as $\mathbf{F}_{\text{ROMaug}}(\cdot)$, can be expressed as

$$\dot{\mathbf{x}}(t) = \mathbf{F}_{\text{ROM}}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t)) + \Delta \mathbf{F}_{\text{ROM}}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t), \mathbf{p}), \quad (1)$$

where $\mathbf{F}_{\text{ROM}}(\cdot)$ represents the original nonlinear ROM dynamics. By incorporating $\Delta \mathbf{F}_{\text{ROM}}(\cdot)$, the adapted model more accurately captures the behavior of the plant. Here, $\mathbf{x}(t)$, $\mathbf{u}(t)$, and $\mathbf{d}(t)$ represent the continuous system states, control variables, 105 and external disturbances, respectively. The correction $\Delta \mathbf{F}_{\text{ROM}}$ depends on these same quantities, but also on free parameters \mathbf{p} that are learnt based on data.

2.1 Reduced order model

A simplified wind turbine model with only three degrees of freedom (drivetrain angular speed, and tower fore-aft and side-side deflections) is considered to represent the ROM dynamics $\mathbf{F}_{\text{ROM}}(\cdot)$. The incident wind V_w induces an aerodynamic torque T_Q 110 about the rotor axis and an aerodynamic force F_T along it. The aerodynamic torque directly excites the drivetrain rotational dynamics

$$J_r \dot{\omega} = T_Q - T_g, \quad (2)$$

ignoring mechanical losses, where J_r , ω , and T_g represent the rotor moment of inertia, rotor speed, and generator torque referred to the low-speed shaft, respectively. The aerodynamic force F_T , coupled with the drive-train dynamics, excites oscil- 115 lations in the tower. These can be quantified by using the tower-top deflection in the fore-aft direction $d_{T_{\text{FA}}}$, having dynamics

$$\ddot{d}_{T_{\text{FA}}} = \frac{1}{\mathbb{f}_1} (F_{T_{\text{FA}}} - \mathbb{f}_2 \dot{d}_{T_{\text{FA}}} - \mathbb{f}_3 d_{T_{\text{FA}}}), \quad (3)$$

and the tower-top deflection in the side-side direction $d_{T_{\text{SS}}}$, having dynamics

$$\ddot{d}_{T_{\text{SS}}} = \frac{1}{\mathbb{s}_1} (F_{T_{\text{SS}}} - \mathbb{s}_2 \dot{d}_{T_{\text{SS}}} - \mathbb{s}_3 d_{T_{\text{SS}}} - \mathbb{s}_4 T_g). \quad (4)$$

120 These result in cyclic stresses $\sigma_{\text{FA}}(t)$ and $\sigma_{\text{SS}}(t)$, respectively, at tower base. Here, $F_{T_{\text{FA}}}$ and $F_{T_{\text{SS}}}$ represent the rotor orthogonal (thrust) and rotor in-plane (side-side force) components of the aerodynamic force F_T , whereas \mathbb{f}_{1-3} and \mathbb{s}_{1-4}

are model parameters. The aerodynamic torque $T_Q(\omega, \beta_b, (V_w - \dot{d}_{T_{FA}}))$, and force components $F_{T_{FA}}(\omega, \beta_b, (V_w - \dot{d}_{T_{FA}}))$ and $F_{T_{SS}}(\omega, \beta_b, (V_w - \dot{d}_{T_{FA}}))$ introduce non-linearities in the model.

The turbine model has two control variables: the commanded generator torque T_{gc} , and the commanded blade pitch angle β_c . The dynamics of the generator is given as

$$\dot{T}_g = \frac{1}{\mathfrak{g}_1} (T_{gc} - T_g), \quad (5)$$

where coefficient \mathfrak{g}_1 represents the time-constant of the first-order dynamic model. The pitch dynamics is modeled as

$$\ddot{\beta}_b = -\mathfrak{b}_1 \dot{\beta}_b - \mathfrak{b}_2 (\beta_b - \beta_c). \quad (6)$$

Here, β_b represents the effective collective blade pitch angle and the coefficients \mathfrak{b}_{1-2} are model parameters representing properties of the pitch system.

The reduced-order wind turbine model consists of eight system states,

$$\mathbf{x} = (\omega, d_{T_{FA}}, \dot{d}_{T_{FA}}, d_{T_{SS}}, \dot{d}_{T_{SS}}, \beta_b, \dot{\beta}_b, T_g), \quad (7)$$

and two control input variables

$$\mathbf{u} = (\beta_c, T_{gc}). \quad (8)$$

The wind speed V_w is considered as a disturbance input to the model, i.e.

$$\mathbf{d} = (V_w). \quad (9)$$

2.2 Data-driven correction

With advancements in computing technologies, machine learning techniques – including supervised and unsupervised learning – have become highly effective for a wide range of data-driven system identification tasks. A key advantage of supervised learning over unsupervised learning is its access to both input states and their corresponding target states during training. This allows the model to generalize and predict system behavior for previously unseen input combinations. A particularly effective approach for supervised learning in this context is based on the training of a neural network (NN). NNs can approximate arbitrarily complex functions while also providing gradients, making them especially well-suited for integration within an optimal control framework.

In this work, the NN is designed to establish a static mapping from inputs to outputs. The structure of an NN consists of multiple hidden layers and a single output layer, where each layer contains a certain number of neurons. These neurons are defined by weights, biases, and activation functions. Within a given layer, input data is first processed by applying the corresponding weights and biases to each neuron. The transformed data is then passed on to the next layer based on the selected activation function. The choice of activation function plays a crucial role in shaping how information flows through the network.



A feed-forward NN with one hidden layer and one output layer is considered in this work. The static mapping of the input features \mathbf{x}_{NN} to the output features \mathbf{y}_{NN} for the considered NN can be written as

$$\mathbf{y}_{NN} = \mathbf{f}_y^{\text{act}}(\mathbf{W}_y^T \mathbf{f}_h^{\text{act}}(\mathbf{W}_h^T \mathbf{x}_{NN} + \mathbf{b}_h) + \mathbf{b}_y). \quad (10)$$

Here, \mathbf{W} , \mathbf{b} , and $\mathbf{f}^{\text{act}}(\cdot)$ represent the weights, biases, and activation functions, respectively, where the subscript h denotes the hidden layer and the subscript y denotes the output layer.

Before training the NN, the generated dataset is split into training and testing sets. Moreover, every feature in the input and the target set is normalized by subtracting the minimum of the feature vector from the feature vector itself, and then dividing the result by the range (difference of maximum and minimum), i.e.:

$$\mathbf{y}_{NN} = \frac{\mathbf{y}_{NN} - \mathbf{y}_{NN}^{\min}}{\mathbf{y}_{NN}^{\max} - \mathbf{y}_{NN}^{\min}}. \quad (11)$$

Furthermore, the normalized datasets are shuffled to reduce the possible clustering of conditions that might create biases. After pre-processing the training set, the parameters of the NN,

$$\mathbf{p} = (\mathbf{W}_h, \mathbf{b}_h, \mathbf{W}_y, \mathbf{b}_y), \quad (12)$$

are computed using the Levenberg-Marquardt algorithm to minimize the sum of squares of error between the predicted output and target output.

The accuracy of NN predictions heavily depends on the quality of its training process, which requires a comprehensive dataset of inputs and corresponding target outputs. Here data is obtained using a high-fidelity simulation model, using both standard measurements and estimates derived from them. Figure 1 illustrates the proposed approach of acquiring input and target data for NN training, where the solid blue circles represent the plant states $\mathbf{x}_i^{\text{Plant}}$ at a given time instant i . To generate training data, the ROM is initialized with each Plant state sample, setting $\mathbf{x}_i^{\text{ROM}} = \mathbf{x}_i^{\text{Plant}}$. The ROM then predicts the next state $\mathbf{x}_{i+1}^{\text{ROM}}$, based on the applied control inputs $\mathbf{u}_i^{\text{Plant}}$ and the influence of external disturbance $\mathbf{d}_i^{\text{Plant}}$.

The corresponding error between the derivatives of the ROM model and the plant,

$$\mathbf{e}_{i+1} = \dot{\mathbf{x}}_{i+1}^{\text{Plant}} - \dot{\mathbf{x}}_{i+1}^{\text{ROM}}, \quad (13)$$

constitutes the target dataset. The derivatives are calculated using the current state $\mathbf{x}_i^{\text{Plant}}$ at time t_i and corresponding next states $\mathbf{x}_{i+1}^{\text{Plant}}$ and $\mathbf{x}_{i+1}^{\text{ROM}}$ at time t_{i+1} via finite differences, i.e. $(\mathbf{x}_{i+1} - \mathbf{x}_i)/(t_{i+1} - t_i)$. The set of the current states, control inputs, and disturbances applied to the plant, constitute the input dataset $(\mathbf{x}_i^{\text{Plant}}, \mathbf{u}_i^{\text{Plant}}, \mathbf{d}_i^{\text{Plant}})$ for the i th time sample. This simple temporal discretization, although of limited numerical accuracy, is deemed sufficient for the present application, where modeling errors are typically larger than the numerical ones.

3 ENMPC formulation

The ENMPC is formulated based on the augmented internal model ROM_{aug} , as expressed by Eq. (1), to predict the system states over a short future horizon T_{horizon} . The discretization of time-continuous variables is performed over control time steps

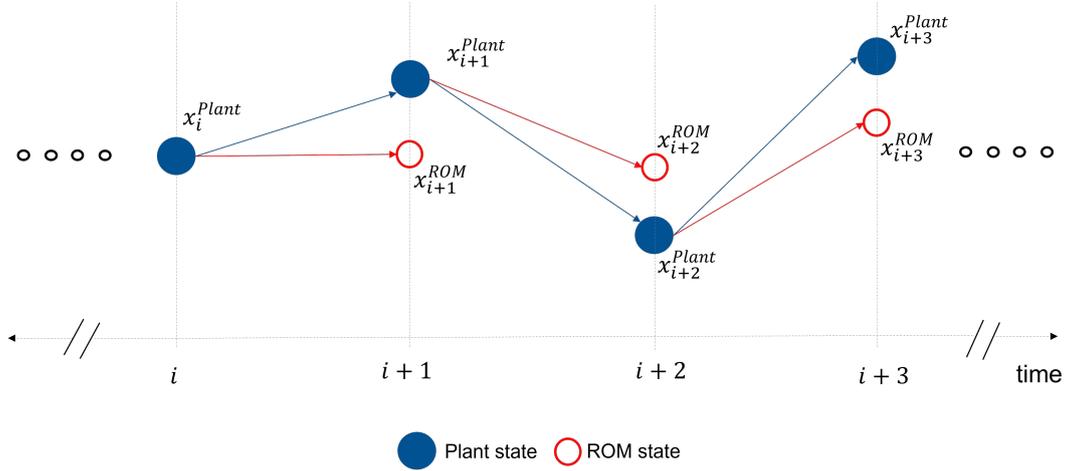


Figure 1. Data acquisition for model adaptation.

N_u in the prediction horizon. The duration of one control interval can thus be denoted as $T_{\text{ctrl}} = T_{\text{horizon}}/N_u$. These predictions are used to calculate optimal control variables by optimizing a desired realistic and meaningful economic objective function (here chosen as profit, with the goal of balancing revenue and costs). Moreover, the controller should keep the plant states and applied inputs within feasible ranges, which are specified as optimization constraints.

185 The proposed ENMPC optimization problem can be formulated as

$$\min_{\mathbf{u}, \xi} -(J_{\text{generation}}^{\text{Power}})^2 + (J_{\text{tower fatigue}}^{\text{FA}})^2 + (J_{\text{tower fatigue}}^{\text{SS}})^2 + \int_{t_0}^{t_0 + T_{\text{horizon}}} (W_1^\xi \xi_1^2 + W_2^\xi \xi_2^2) dt, \quad (14a)$$

subject to

$$\dot{\mathbf{x}} = \mathbf{F}_{\text{ROM}_{\text{aug}}}(\mathbf{x}, \mathbf{u}, \mathbf{d}, \mathbf{p}), \quad (14b)$$

190 $\underline{\mathbf{x}}_c \leq \mathbf{x}_c \leq \overline{\mathbf{x}}_c,$ (14c)

$$\underline{\mathbf{u}} \leq \mathbf{u} \leq \overline{\mathbf{u}}, \quad (14d)$$

$$\underline{\xi} \leq \xi \leq \overline{\xi}, \quad (14e)$$

195 and

$$\underline{\dot{T}}_g \leq \dot{T}_g \leq \overline{\dot{T}}_g. \quad (14f)$$

The optimization variables are the control variables $\mathbf{u} = (\beta_c, T_{gc})$ and the slack variables $\boldsymbol{\xi} = (\xi_1, \xi_2)$. The purpose of introducing slack variables is to achieve recursive feasibility of the MPC optimization problem in the presence of model uncertainties and system perturbations (S. Gros, 2013). Hence, in the present formulation, the state variable ω and the wind turbine electrical
 200 power output

$$P_{\text{gen}} = \eta_{\text{gen}} \omega T_g \quad (15)$$

are augmented using the bounded slack variables ξ_1 and ξ_2 , respectively, where η_{gen} denotes the drivetrain conversion efficiency. This approach is used because ω , despite the smoothing effect provided by the large rotor inertia, is subject to wind perturbations and model errors that affect the economic MPC problem (S. Gros, 2013). The modified set of states \mathbf{x}_c can be
 205 expressed as

$$\mathbf{x}_c = (\omega + \xi_1, d_{T_{FA}}, \dot{d}_{T_{FA}}, d_{T_{SS}}, \dot{d}_{T_{SS}}, \beta_b, \dot{\beta}_b, T_g, \eta_{\text{gen}} \omega T_g + \xi_2). \quad (16)$$

3.1 Optimization objective

The optimization objective aims to maximize the generated profit by balancing the revenue accrued from wind power generation and cost incurred due to fatigue damage. The cost of fatigue damage should be expressed by appropriate models, which will
 210 differ depending on the component; for example, it is reasonable to assume that the effects of cyclic loading on a pitch bearing will in general be very different from the ones on a gearbox. Here, for the lack of specific models and relevant data, we simply consider the tower as an exemplary component that is often fatigue critical. It is clear that this is only an academic example, and more realistic scenarios could be readily developed by using the same methodology for multiple components, by using dedicated component-specific cost models.

215 The wind power generation is maximized by considering the aerodynamic power capture

$$J_{\text{generation}}^{\text{Power}} = w_P \int_{t_0}^{t_{\text{end}}} (\omega T_Q(\omega, \beta_b, (V_w - \dot{d}_{T_{FA}}))) dt, \quad (17)$$

where w_P denotes the revenue rate for providing electricity to the grid. It should be noted that even though revenue is accrued based on the overall electrical power generation, in this work the aerodynamic power is maximized. This is to avoid the greedy extraction of rotor kinetic energy by MPC (referred to as “turnpike effect” in S. Gros (2013)).

220 The tower cyclic fatigue damage is minimized by a direct penalization of fatigue via the PORFC approach. The PORFC algorithm uses a pre-processing step to identify fatigue cycles for a given set of stress samples $\boldsymbol{\sigma}$, and splits the respective cyclic fatigue damages over the contributing samples. The output of the pre-processing step is the PORFC mean parameters $\boldsymbol{\sigma}_{m,c}^{\text{PORFC}}$ and PORFC weight parameters $\boldsymbol{\sigma}_{w,c}^{\text{PORFC}}$ (Loew et al., 2021).

A Python script to extract the PORFC parameters for a given set of stress trajectories is shared with this work (Anand and
 225 Bottasso, 2025). This allows for a detailed understanding of the novel formulation and its adaptability and usage for economic MPC. The script performs standard rainflow counting to identify cycle characteristics and extract PORFC parameters. These parameters are then used to reformulate the cycle amplitudes and weights over stress samples in a continuous manner.



The cost of tower fatigue damage $J_{\text{tower fatigue}}^{\text{FA}}$, due to tower root fore-aft cyclic stress σ_{FA} , is formulated as

$$J_{\text{tower fatigue}}^{\text{FA}} = \frac{1}{T_{\text{ctrl}}} \int_{t_0}^{t_{\text{end}}} \sum_{c=1}^2 J_{c, \sigma_{\text{FA}}}^{\text{PORFC}}(\sigma_{\text{FA}}, \sigma_{\text{FA} m, c}^{\text{PORFC}}, \sigma_{\text{FA} w, c}^{\text{PORFC}}) dt, \quad (18a)$$

230

$$J_{c, \sigma_{\text{FA}}}^{\text{PORFC}} = \sigma_{\text{FA} w, c}^{\text{PORFC}} a_m \sigma_{\text{FA} eq, c}^m, \quad (18b)$$

$$\sigma_{\text{FA} eq, c} = |\sigma_{\text{FA}} - \sigma_{\text{FA} m, c}^{\text{PORFC}}| \frac{R_m}{R_m - \sigma_{\text{FA} m, c}}. \quad (18c)$$

Here, a_m denotes the capital cost of the component and is determined from the initial capital expenditure of the machine (see also Loew et al. (2023) for details), R_m denotes the ultimate tensile strength of the material, and m represent the positive exponent derived from the material S-N characteristic.

The tower cyclic fatigue damage $J_{\text{tower fatigue}}^{\text{SS}}$, due to tower root side-side cyclic stress σ_{SS} , is formulated in a similar manner.

Although the optimization objective, shown in Eq. (14a), separately considers the costs of tower fore-aft and side-side fatigue, the final evaluation is based on the projected total cost. In addition to the fact that fore-aft and side-side components depend on wind direction, which is not constant, this also ensures that any potential increase or decrease in tower side-side stress oscillations, resulting from control actions aimed at minimizing tower fore-aft stress (and vice-versa), is properly accounted for.

To obtain the cost of cyclic fatigue damage for each projection at a given azimuth direction, the following steps are taken. First, rainflow counting is performed on the projected stress trajectory. Then, the Goodman equation is applied for mean stress correction. The damage cost of each stress cycle is determined using the S-N curve of the material of the tower and the component cost. Finally, the Miner-Palmgren algorithm is used to sum the costs of individual cycles and obtain the total cost (refer to Loew et al. (2023) for a detailed formulation).

This approach – here illustrated for the tower – is readily generalized to other components. Once cyclic fatigue is assessed on each prediction horizon for each component of interest, a dedicated model could provide failure rates and/or maintenance activities resulting from such loading, in turn generating the associated costs, which would be included in the optimization merit function.

3.2 Optimization constraints

The ENMPC optimization problem is subjected to the system dynamics of the augmented plant model $\text{ROM}_{\text{aug}}(\cdot)$, as shown in Eq. (14b), to the inequality constraints on modified system states, as shown in Eq. (14c), and to the box constraints on control and slack variables, as shown in Eq. (14d) and Eq. (14e). Furthermore, the rate of change of generator torque \dot{T}_g is also subjected to an inequality constraint, as shown in Eq. (14f), to reduce the torque travel as well as the fatigue in the wind turbine drive-train.



3.3 State estimator

The controller-internal model is initialized using the currently measured initial states \mathbf{x}_0 . However, not all system states of the internal model can actually be measured directly on the plant using standard on-board sensors. For instance, both the tower-top deflection and velocity states ($d_{T_{FA}}, \dot{d}_{T_{FA}}, d_{T_{SS}}, \dot{d}_{T_{SS}}$) for ROM and ROM_{aug} can not be measured directly on a real turbine. Only the rotor speed ω , the blade pitch angle β_b , and the tower-top accelerations $\ddot{d}_{T_{FA}}$ and $\ddot{d}_{T_{SS}}$ can be measured by onboard sensors, whereas the remaining states need to be estimated using measured quantities. Furthermore, as the controller-internal model is only a reduced representation of the plant, the initial values measured directly on the plant may not be suitable for initializing the ENMPC. As a consequence, a state estimator is additionally required to provide initial value estimates \mathbf{x}_{est} of the system states for the ENMPC internal model, using the available measurements from the plant.

A classical approach for state estimation is the Kalman filter, also widely used for wind turbine control (Bottasso and Croce, 2009; Ritter, 2020). However, due to the nonlinear nature of the system and the need to enforce constraints on both stage and terminal states, here we instead adopt a moving horizon estimator (MHE). A detailed comparison between optimization-based state estimation techniques based on MHE and Kalman filters has been presented in Loew and Bottasso (2022). The MHE formulation used in this study builds upon the approach discussed in Anand et al. (2022).

MHE utilizes the system information from the plant over a finite past duration (specified using the MHE horizon length $T_{horizon,est}$), to calculate the initial state estimates $\mathbf{x}_{est}(t_0)$ for the current ENMPC step. The MHE optimization problem aims to minimize the sum of the deviation of the current estimated output \mathbf{y}_{est} from the measurement values \mathbf{y}_{meas} , with the deviation of the current state estimates \mathbf{x}_{est} from the previous state estimates $\mathbf{x}_{est,prev}$ (to ensure a smooth estimator output over consecutive MHE steps), and with the noise variable $\bar{\nu}$ (S. Gros, 2013; Huang et al., 2010).

The objective function is given as

$$\min_{\bar{\nu}} \int_{t_0 - T_{horizon,est}}^{t_0} (\|\mathbf{y}_{est} - \mathbf{y}_{meas}\|_{\mathbf{W}_{meas}}^2 + \|\mathbf{x}_{est} - \mathbf{x}_{est,prev}\|_{\mathbf{W}_{prev}}^2 + \|\bar{\nu}\|_{\mathbf{W}_{\bar{\nu}}}^2) dt, \quad (19)$$

where $\mathbf{y}_{est} = (\mathbf{x}_{est}, \ddot{d}_{T_{FA,est}}, \ddot{d}_{T_{SS,est}})$ and $\mathbf{y}_{meas} = (\mathbf{x}_{meas}, \ddot{d}_{T_{FA,meas}}, \ddot{d}_{T_{SS,meas}})$. The estimated tower-top fore-aft acceleration $\ddot{d}_{T_{FA,est}}$ and side-side acceleration $\ddot{d}_{T_{SS,est}}$ are obtained using the nonlinear output equation expressed by Eq. (3) and Eq. (4), respectively. The measured tower-top fore-aft acceleration $\ddot{d}_{T_{FA,meas}}$ and the side-side acceleration $\ddot{d}_{T_{SS,meas}}$ are obtained from the plant as a result of standard sensor measurements. The measured tower-top velocity and deflection in both the fore-aft and side-side directions are obtained by numerical integration of the tower-top acceleration and velocity, respectively. The diagonal weighting matrices \mathbf{W}_{meas} , \mathbf{W}_{prev} , and $\mathbf{W}_{\bar{\nu}}$ are obtained by a trial and error tuning, such that a satisfactory performance is achieved.

The optimization problem is subjected to the estimator system dynamics

$$\dot{\mathbf{x}}_{est} = \mathbf{F}_{est}(\mathbf{x}_{est}(t), \mathbf{d}_{est}(t)) + \boldsymbol{\nu}(t), \quad (20)$$

where $\mathbf{d}_{est}(t) = (V_w(t), \beta_b(t), T_g(t))$ are the disturbance inputs to the system, which are already set by the ENMPC and are hence fixed for the current MHE step. Here, \mathbf{x}_{est} represents the estimator system states, corresponding to the wind turbine



system states \mathbf{x} , as discussed in Sect. 2.1. Moreover, $\mathbf{F}_{\text{est}}(\cdot)$ represents the system of ODEs for wind turbine dynamics discussed
290 in Sect. 2.1.

After the execution of an MHE step, the terminal state at the end of the MHE horizon becomes the initial state at the beginning of the ENMPC prediction horizon, i.e. $\mathbf{x}_0 = \mathbf{x}(t_0) = \mathbf{x}_{\text{est}}(t_0)$.

3.4 Disturbance estimator

The wind speed V_w is considered as a disturbance input to the ENMPC formulation, and needs to be estimated over the
295 prediction horizon T_{horizon} of the controller. This work considers a simple rotor effective wind speed (REWS) estimator, based on the approach discussed in Soltani et al. (2013). The REWS estimator utilizes the drivetrain dynamics, see Eq. (2), to estimate the aerodynamic torque

$$T_{Q_{\text{est}}} = \frac{P_{Q_{\text{est}}}}{\omega_{\text{meas}}} \approx \frac{P_{\text{gen}_{\text{meas}}} + J_r \dot{\omega}}{\omega_{\text{meas}}}, \quad (21)$$

using measured generator power $P_{\text{gen}_{\text{meas}}}$ and rotor speed ω_{meas} . The rate of change of rotor speed $\dot{\omega}$ is computed by finite dif-
300 ference from ω_{meas} . The estimated aerodynamic torque $T_{Q_{\text{est}}}$ is then equated to the aerodynamic torque $T_Q(\omega_{\text{meas}}, \beta_{b_{\text{meas}}}, V_{w_{\text{est}}})$ described in Sect. 2.1, to estimate wind speed $V_{w_{\text{est}}}$, for the measured pitch angle $\beta_{b_{\text{meas}}}$ and ω_{meas} .

4 Results and discussions

4.1 Case study

The ability of the adapted model to accurately predict the plant states depends directly on the precision of the data-driven
305 corrections. Additionally, to determine the extent of model adaptation required, it is necessary to assess how reducing the model mismatch impacts the closed-loop performance of the controller. To answer these questions, we consider a plant represented by the NREL 5 MW reference wind turbine (Jonkman et al., 2009), which is modeled using OpenFAST (Bonnie Jonkman et al., 2022), a widely used tool for simulating wind turbine dynamics. The plant model includes the first and second flapwise bending modes and the first edgewise bending mode for each of the three blades. Additionally, it also includes the first and
310 second tower bending modes in both fore-aft and side-side directions, as well as the torsional flexibility of the drivetrain and the generator DOF.

The model, which incorporates pitch and torque actuators but excludes the yaw mechanism, consists of thirty-three system states. There are eight states for the tower dynamics, eighteen for the blades (six per blade), two for drive-shaft torsion, two for rotor rotation, two for collective blade pitch actuation, and one for generator torque actuation.

315 The fixed model parameters of the corresponding ROM, namely \mathbb{f}_{1-3} , \mathbb{s}_{1-4} , \mathbb{g}_1 , and \mathbb{b}_{1-2} , are derived from the NREL report describing the 5 MW reference wind turbine (Jonkman et al., 2009).

4.1.1 Data generation

The performance of a NN heavily depends on the quality of its training, which requires an exhaustive dataset that encompasses a wide range of operating conditions. To generate a comprehensive training dataset, the OpenFAST model is simulated in turbulent wind conditions using the baseline controller provided in the OpenFAST package. Full-field turbulent wind inputs are generated using TurbSim (Jonkman, 2009), considering the class-B normal turbulence model. Simulations are performed for wind speeds ranging from cut-in to cut-out, in 1 ms^{-1} steps. For each wind speed, six different turbulent wind seed simulations are conducted.

Figure 2 presents the Pearson correlation coefficients of the states (\mathbf{x}), inputs (\mathbf{u}), and disturbances (\mathbf{d}) of the ROM with the corresponding state errors (refer to Sect. 2.2), computed over the entire dataset. The Pearson correlation coefficient quantifies the linear relationship between two variables (Nafis Faizi and Yasir Alvi, 2023). Each entry in the table of Fig. 2 displays the absolute value of the coefficient, where a higher magnitude indicates a stronger correlation. The same relationship is visualized through a color gradient map, where darker shades represent stronger correlations and lighter shades indicate weaker ones.

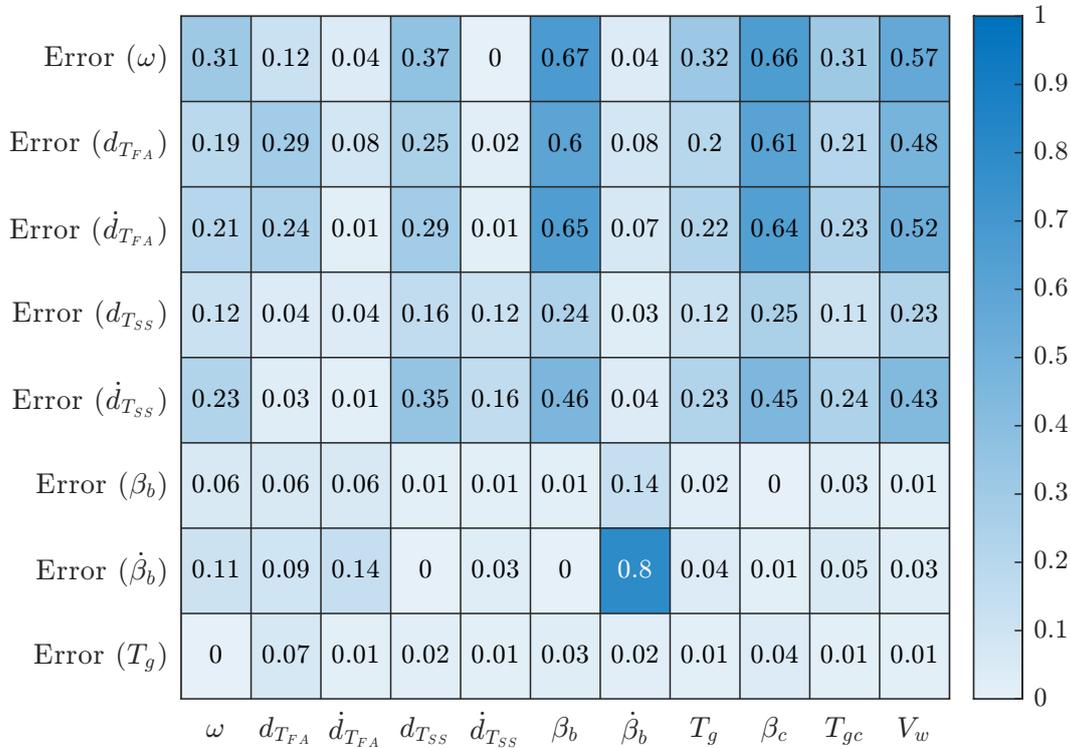


Figure 2. Correlation coefficient of states, inputs, and disturbance of the controller-internal model (as columns) with the error in states (as rows) for the generated dataset, rounded off to two significant digits. The number in each cell denotes the absolute value of the Pearson correlation coefficient, while the cell color visually denotes the degree of correlation.



Figure 2 shows that all ROM states and inputs significantly influence the state errors, as evidenced by their nonzero correlation coefficients. In particular, the rotor speed (ω) and blade states ($\beta_b, \dot{\beta}_b$) exhibit the strongest correlation with both rotor speed errors and tower state errors. Additionally, the control inputs (β_c and T_{gc}) and wind speed (V_w) strongly affect the mismatch across all states. Furthermore, the nonzero yet distinct correlation magnitudes for blade pitch (β_b) and collective blade control (β_c) highlight the importance of blade dynamics, shown in Eq. (6), in the internal model of the controller. This suggests that, while the ROM captures an approximation of the complex aeroelastic response, it does not fully represent the detailed dynamics present in the plant.

4.1.2 NN training

The proposed NN input set \mathbf{x}_{NN} contains eleven features (eight system states, two control variables, and one disturbance input), and each output set \mathbf{y}_{NN} contains eight features representing errors in the derivatives of states calculated using the plant and ROM states. The selection of input features is guided by the correlation coefficients presented in Fig. 2, which indicate a nonzero correlation between all variables in the input and target sets. The hidden layer of the network consists of twenty neurons, each utilizing a radial basis activation function, while the output layer comprises eight neurons with linear activation functions. The number of neurons in the hidden layer was determined through hyperparameter tuning, ensuring optimal training performance.

Before training the NN, the generated dataset is split into training and testing sets. During wind turbine operation, more data is naturally collected for operating conditions corresponding to more probable wind speed values. To account for this effect in the application of the proposed methodology, the dataset split is performed according to a hypothetical site-specific Weibull distribution. Figure 3 illustrates the distribution of selected input features within the training dataset, which consists of 216,071 sample points. The various subplots show that the dataset covers a broad operational range of the wind turbine, allowing the NN to predict the target variables across a wide range of input conditions. Furthermore, Fig. 3i confirms that the wind speed distribution in the training dataset closely follows the Weibull shape, ensuring a realistic representation of operating conditions.

4.2 Open-loop evaluation

The ability of the augmented internal model to accurately track system states is necessary for ensuring both the optimality and stability of the closed-loop behavior. To evaluate this aspect of the proposed approach, both ROM_{aug} and ROM are simulated using each input combination from the testing dataset (refer to Sect. 4.1.2), with the initial state accordingly set. The final states predicted by both models are then compared to the corresponding plant states, and the prediction error is computed to quantify the accuracy of each model.

Figure 4 presents the Kernel-density plot of the absolute prediction errors for both ROM (solid blue line) and ROM_{aug} (solid green line) across all system states. For each of the subplot, the x -axis represents the absolute error values, while the y -axis shows the probability density estimate (PDE), calculated using a normal kernel function.

In the ROM case, certain states, such as “Tower-top FA defl.” and “Tower-top FA vel.”, exhibit their highest PDE at a nonzero error value, indicating a mismatch in model fidelity. In contrast, in the ROM_{aug} case, this mismatch is corrected

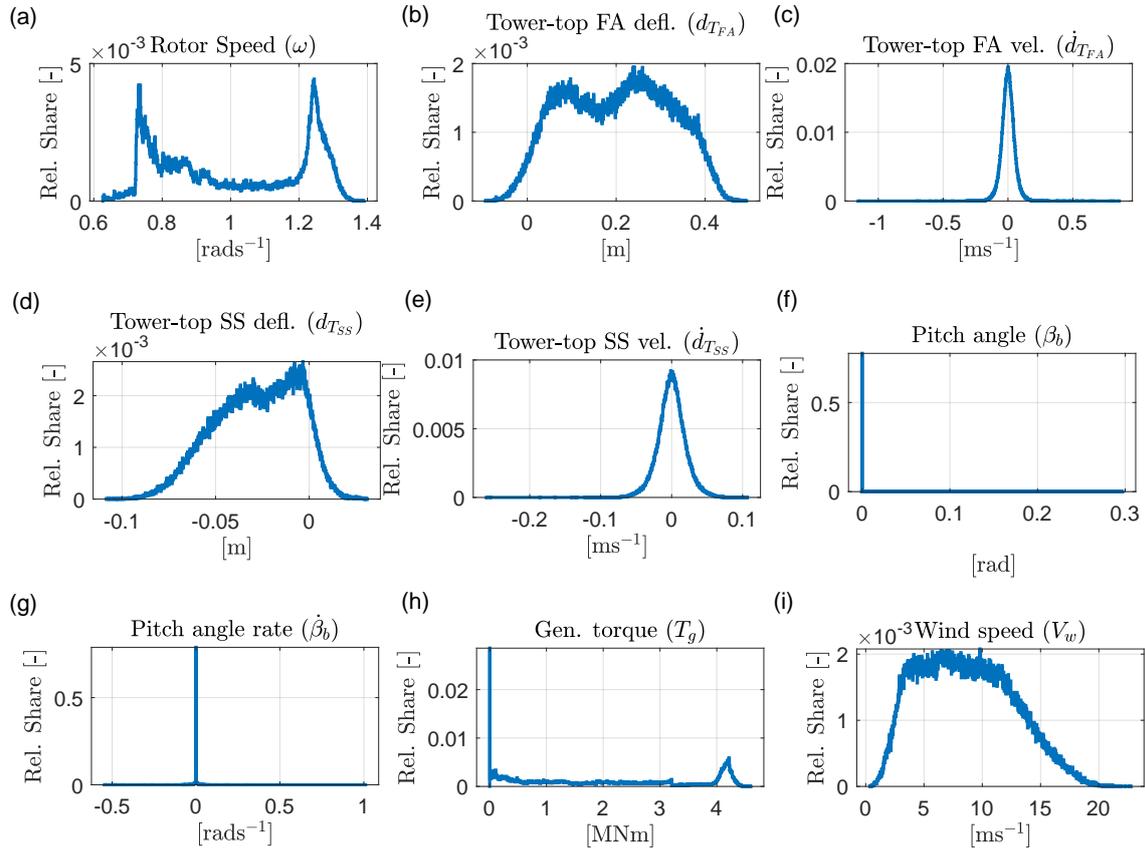


Figure 3. Distribution of different input features in the training set.

using the proposed adaptation scheme. As a result, the PDE peak shifts toward zero, signifying an improved state prediction accuracy. Furthermore, for most system states, the PDE peak magnitude in the ROM_{aug} case is higher than in the ROM case. This indicates a greater concentration of cases where ROM_{aug} achieves lower absolute errors than ROM, further demonstrating the effectiveness of the proposed model augmentation. The figure also shows that the accuracy of some quantities cannot be improved, although their typical errors are always very small.

To quantify the effectiveness of ROM_{aug} in minimizing plant-model mismatch in open-loop, a statistical evaluation was conducted using the mean and standard deviation (STD) of prediction errors. Table 1 presents the percentage reduction in both the average error and standard deviation of errors for ROM_{aug}, relative to ROM, across the test set.

The results show varying degrees of improvement in state prediction, demonstrating that NN-based augmentation reduces not only the average error but also the error spread over different operating conditions. A greater reduction in prediction error is observed for states with higher absolute error magnitudes in Fig. 4. This highlights the effectiveness of weight tuning during the NN training stage, which prioritizes correcting states with significant errors. These findings further emphasize the effectiveness

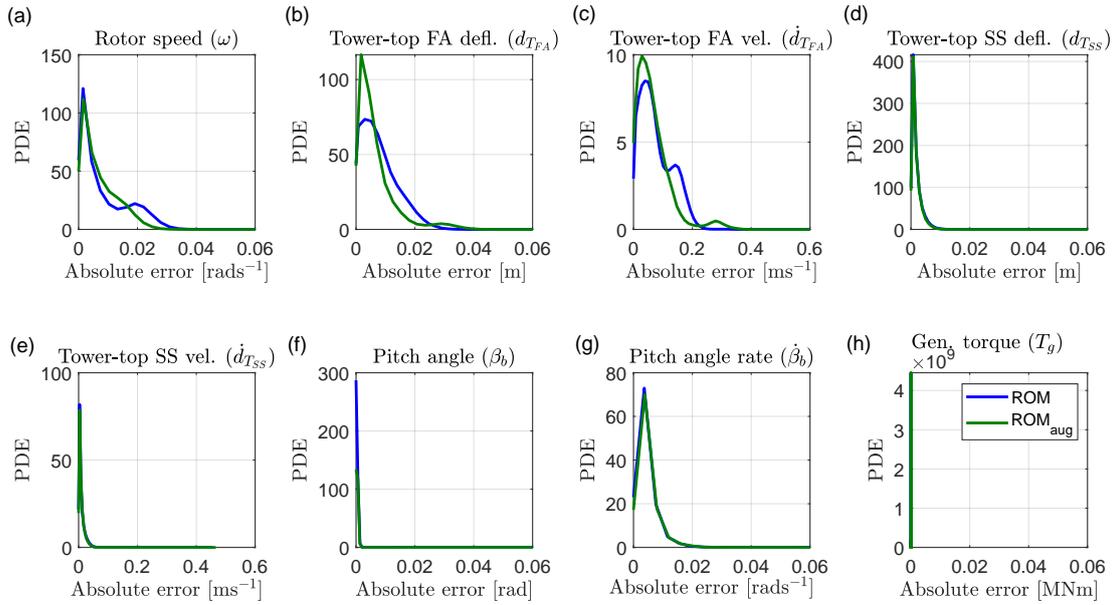


Figure 4. Probability density estimate (PDE) of the open-loop prediction errors for both ROM_{aug} and ROM, shown in green and blue, respectively, evaluated over the test set.

Table 1. Percentage reduction in mean error and standard deviation (STD) error due to model augmentation assessed on the test set.

State	Mean error reduction [%]	STD error reduction [%]
Rotor speed (ω)	25.37	26.51
Tower-top FA defl. ($d_{T_{FA}}$)	20.55	12.57
Tower-top FA vel. ($\dot{d}_{T_{FA}}$)	11.02	9.73
Tower-top SS defl. ($d_{T_{SS}}$)	7.91	8.14
Tower-top SS vel. ($\dot{d}_{T_{SS}}$)	7.76	9.05
Pitch angle (β_b)	5.69	5.96
Pitch angle rate ($\dot{\beta}_b$)	0.08	0.53
Gen. torque (T_g)	0.01	0.00



of the proposed offline data-driven approach in reducing plant-model mismatch, improving the overall accuracy of the internal
375 model.

For a practical application of the model adaptation, it is interesting to quantify the amount of training data that is needed
to achieve a desired level of performance. To this end, multiple data subsets are created from the original training set. Each
subset contains a specified fraction of the data samples from the original training set. Figure 5a illustrates the distribution
of wind speed for multiple subsets, ranging from 45% to 95% of the original training set. The x -axis represents the wind
380 speed magnitude, while the y -axis shows the corresponding relative share. Each subset is used to train a NN with the same
architecture as described in Sect. 4.1.2.

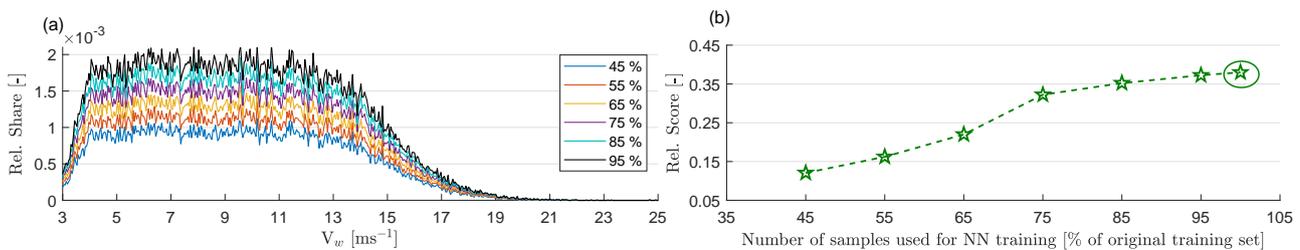


Figure 5. Open-loop evaluation of the proposed model adaptation where the NNs are trained using subsets of the full training set. Relative share of wind speed (V_w) in the NN training set (a). Corresponding performance score of ROM_{aug} relative to the performance score of ROM as a function of test set size (b).

Figure 5b presents the performance of the adapted models evaluated on the same test set. The performance score is calculated by summing the root mean squared error (RMSE) of predictions for all eight states. For ease of comparison, the performance scores of the different ROM_{aug} models are normalized against the performance score of ROM. The y -axis shows the relative
385 values, while the x -axis displays the different ROM_{aug} formulations trained using the data subsets shown in Fig. 5a. Additionally, Fig. 5b includes the normalized performance score of the ROM_{aug} model trained with the full dataset, represented by the green circle.

The results show that, as expected, incorporating more data into the training process provides the NN with additional information, leading to improved performance. However, even a relatively small subset of training data helps reduce the plant-model
390 mismatch. Furthermore, performance begins to level off beyond a certain point, as additional data samples no longer significantly contribute to improving the model. In this study, performance starts to plateau once 85% of the training dataset is used.

4.3 Closed-loop evaluation

Although the prediction errors of the augmented ROM are still non-zero for some operating conditions (refer to the long
395 tail of the PDE plots in Fig. 4), it is useful to understand the impact of plant-model mismatch reduction on closed-loop



performance, as this helps quantify the necessary level of compensation. To investigate this aspect of the formulation, the proposed adaptive economic controller is implemented in closed-loop with the plant. This configuration will be henceforth referred to as ENMPC_{aug}. To assess the impact of the augmentation, the performance of the plant is also evaluated using the same ENMPC based on the original non-augmented ROM, which is referred to as ENMPC.

400 We consider the design load case (DLC) 1.2 at 11 ms^{-1} wind speed. The ENMPC-MHE optimization problem (refer to Sect. 3) is solved via the state-of-the-art Acados framework (Verschuere et al., 2019). The interior-point solver HPIPM is used for solving the underlying quadratic programs (QP) within the nonlinear program (NLP). Several sequential quadratic program (SQP) iterations are carried out at each controller step. The multiple shooting approach is employed with a Newton step length of 1. To address potential numerical issues caused by the highly non-standard formulation produced by PORFC
405 (Loew et al., 2020b), the Hessian matrix is automatically convexified. It is important to note that solving multiple SQPs can improve performance. However, it also increases the computational burden.

The ENMPC and MHE horizon lengths, T_{horizon} and $T_{\text{horizon,est}}$, are both set to 2 seconds, each having 20 discretization steps. This results in a sample time T_{ctrl} of 100 ms for both the controller and the estimator. The sample time of the plant, T_{sim} , is set to 10 ms. The optimal control inputs applied to the plant model are considered as piece-wise constant values over T_{ctrl} .
410 Measurements from the plant are taken every T_{ctrl} .

The closed-loop simulations run for a duration of 10 minutes. Six turbulent wind speed seeds, generated using TurbSim, are used to characterize uncertainties in fatigue damage. The controller uses the estimated wind speed REWS (refer to Sect. 3.4) as input, which remains constant over the prediction horizon of the controller.

The closed-loop control performance is evaluated using the following performance indicators:

- 415
1. Revenue due to power generation, calculated considering a fixed feed-in tariff w_P and turbine electrical power generation $\eta_{\text{gen}} \omega T_g$ (refer to Eq. 17 for details).
 2. Cost due to projected tower base fatigue damage. To calculate the projected damage, the tower fore-aft $\sigma_{\text{FA}}(t)$ and tower side-side oscillations $\sigma_{\text{SS}}(t)$ are first projected along the various azimuth directions at tower base. Next, the fatigue damage cost is computed for each of these projections, as discussed in Sect. 3.1. Finally, the maximum cost across all
420 projections is selected.
 3. Profit, calculated as a difference of revenue and cost.
 4. Pitch travel, showing the total degrees that the blades traveled for a given control formulation. This can be considered as a proxy for the usage of pitch actuators, which may be prone to wear and tear, calling for extra maintenance.
 5. Torque travel, showing the total amount of torque that the generator had to apply. This can be considered as a proxy for
425 the usage of the turbine drivetrain, leading to wear and tear of bearings and gearbox. Additionally, it also serves as a proxy for switching-related damage in the power electronic converters.



4.3.1 Economic performance of the controller

Figure 6 presents the performance indicators calculated over a 10 minute simulation, with results averaged across different seeds. The set of bars in the left and right parts of the figure represent the performance for the ENMPC and ENMPC_{aug} formulations, respectively. The color of the bars corresponds to different performance indicators. The results for ENMPC_{aug} are normalized with the ENMPC ones to facilitate comparison, and the *y*-axis shows the relative values. The black numbers above each bar indicate the relative cumulative value compared to the ENMPC case. Additionally, the numbers on the face of each bar, shown in purple text, represent the absolute values.

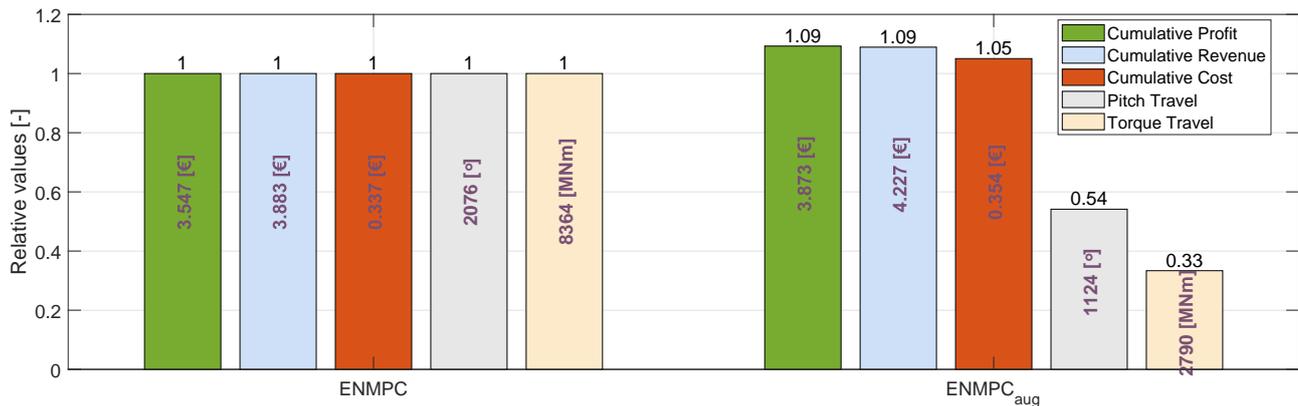


Figure 6. Performance indicators for closed loop simulations using controllers employing the baseline ROM (ENMPC) and the augmented ROM (ENMPC_{aug}) as internal models. The black numbers on the top of each bar denote the corresponding relative cumulative values. The purple text on the face of each bar denote the absolute cumulative values.

The plots in Fig. 6 show that ENMPC_{aug} results in 9% higher economic profit than ENMPC. This improvement is due to the more accurate estimation of revenue and cost within ENMPC_{aug}, which is made possible by a better prediction of the system states. The higher profit is directly attributed to an increased revenue, with a slight increase in cost. Since the absolute economic value of revenue exceeds that of cost (as shown by the purple numbers on the face of the bars), ENMPC_{aug} effectively balances the two, leading to a higher overall economic profit.

In contrast, for the ENMPC case, while the controller aims to maximize economic profit, the plant-model mismatch leads to control actions that are not economically optimal. As a result, the controller struggles to accurately estimate and balance revenue and cost, ultimately resulting in a lower economic profit.

Furthermore, since the future predictions within the controller more closely match the actual evolution of the plant in the ENMPC_{aug} formulation, the controller requires less frequent control actions. This is reflected in the significantly smaller pitch and torque travel compared to the ENMPC case.



445 As a result, reducing plant-model mismatch to improve system state predictions, not only enhances performance but also leads to a substantial reduction in actuator usage. This, in turn, results in lower maintenance costs for both the actuators and the drivetrain. This effect was however not quantified for a lack of data and specific reliable models.

4.3.2 Benefits under different wind inputs

450 Advanced wind turbine control formulations require the current wind speed as an input. Improved foresight of the wind speed enables the ENMPC to determine the most suitable control actions.

Wind speed can be gauged using a simple wind-speed estimator, as utilized in this work (refer to Sect. 3.4), or through advanced estimators as discussed in Soltani et al. (2013). Alternatively, the current wind speed can be measured directly, either using a nacelle-mounted anemometer or a light detection and ranging (LiDAR) device. LiDARs not only provide real-time wind speed data but also offer short-term wind speed previews. This capability makes LiDARs an ideal complement to MPC-
455 based wind turbine control, where the MPC optimization uses system states derived from wind speed forecasts over the MPC prediction horizon (Loew and Bottasso, 2022; Canet et al., 2021).

Here, the benefits of the proposed model adaptation for reducing plant-model mismatch under different wind input estimates are assessed. Figure 7 presents various wind profiles used as input to the controller.

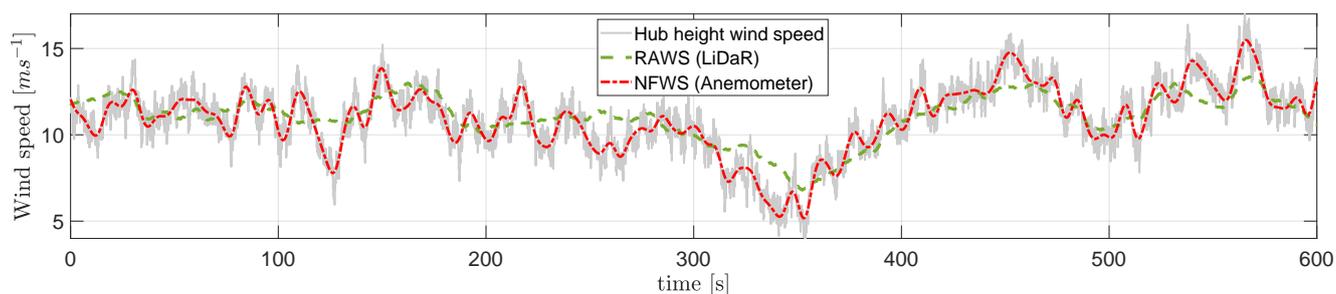


Figure 7. Exemplary wind profiles used as disturbance inputs to the ENMPC formulations.

The gray curve represents the longitudinal component of the full wind field generated using TurbSim at the turbine hub height
460 for a 10 minute duration. This scenario is labeled as “Hub height wind speed”. This is only shown to illustrate wind fluctuations, but it is not used as input to the controller. The red curve corresponds to the nacelle filtered wind speed (NFWS), which is a rough approximation of what an anemometer mounted on the nacelle would measure. Typically, anemometer measurements are subjected to various disturbances, including – among others – flow distortion due to proximity of the nacelle (a bluff body), periodic effects caused by blade passage, interference due to the wake, sensor noise. For the lack of models of these complex
465 phenomena, here we have simply filtered the wind field at the nacelle location using a standard bandpass filter. This case is referred to as “Anemometer”. The green curve shows the rotor averaged wind speed (RAWS). The RAWS is a much simplified



representation of what a nacelle-mounted scanning LiDaR system with discrete scanning and spatial averaging would estimate under a frozen turbulence hypothesis, i.e. with a purely rigid transport of the flow from the measurement volume to the rotor disk (Loew and Bottasso, 2022). This case is referred to as “LiDaR”.

470 It can be observed that the anemometer-measured wind speed follows the highly turbulent fluctuations of the wind speed at hub height. Additionally, the RAWs shows even less dynamic variation in the wind speed but successfully captures the long-term wind speed trends. Although these speeds are only very rough approximations of the wind that could be actually measured on a wind turbine, they still capture a range of situations from point-wise exact values to spatial and temporal averages.

Figure 8 presents the results of the closed-loop control formulations, considering different wind speed estimates. For each 475 wind speed input scenario, two sets of bars are displayed: the first set represents the ENMPC case, followed by the second set for the ENMPC_{aug} case. Within each set, the five performance indicators are shown as five bars, each distinguished by different face colors. The various sets of bars correspond to the following situations:

- The first and second sets show results when the controllers use REWS, described in Sect. 3.4, as disturbance input and the current wind speed is held constant over the prediction horizon. These results are the same as discussed in Fig. 6, and 480 are shown here again for ease of comparison.
- The third and the fourth sets show results when the controllers use NFWS (Anemometer) as disturbance input, and the current wind speed is held constant over the prediction horizon.
- The fifth and the sixth sets show results when the controllers use LiDaR measurements as input, and the current wind speed is held constant over the prediction horizon.
- The seventh and the eighth sets show results when the controllers use LiDaR measurements as input, and a perfect 485 preview of wind condition over the prediction horizon is considered. This case has been labeled as LiDaR-PF.

The results are normalized with respect to the ENMPC case, which is shown in the first set. The black text above each bar represents the relative cumulative values. To aid interpretation, the results from different wind input scenarios are highlighted with distinct background colors. The red background corresponds to the Anemometer scenario, the light green shade represents 490 the LiDaR scenario, and the dark green background indicates the LiDaR-PF scenario.

The results show that better information about the wind disturbance input within the controller leads to increased economic profit for both ENMPC and ENMPC_{aug}. As expected, the Anemometer scenario, providing only a disturbed point-wise wind measurement, results in the lowest economic profit, while the LiDaR-PF scenario yields the highest economic profit.

Furthermore, for all wind input scenarios, the proposed model adaptation results in higher economic profit and reduced 495 pitch and torque travel. The magnitude of the performance improvement varies across the different wind input scenarios. For example, ENMPC_{aug} achieves a 7% higher economic profit in the Anemometer scenario and a 30% increase in the LiDaR scenario, compared to the corresponding ENMPC formulation. It is noteworthy that the economic performance of ENMPC_{aug} in the LiDaR scenario is only slightly worse than that in the LiDaR-PF scenario. This suggests that, for this case study,

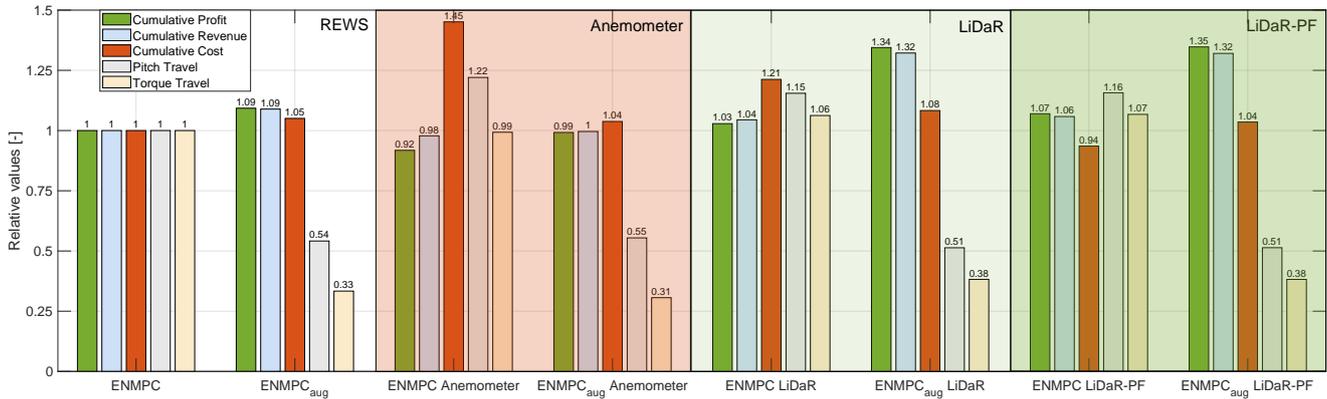


Figure 8. Performance indicators for closed loop simulations using ENMPC and ENMPC_{aug}. The background colors denote different wind input scenarios: white for REWS, red for Anemometer, light green for LiDaR, and dark green for LiDaR-PF. For a given background color, the left and the right columns show ENMPC and ENMPC_{aug} results, respectively. The black numbers on the top of each bar denote the corresponding relative cumulative values.

a perfect wind preview does not significantly pay off, and a simple constant speed estimate is sufficient. However, a more
 500 general conclusion can only be drawn by evaluating the two scenarios over a wider range of turbine inflow conditions.

4.3.3 Computational performance

An economic controller is real-time feasible if the computational time required to generate the optimal control actions is less
 than the sample time of the plant. The length of the prediction horizon, along with the nature of the underlying internal model
 and the optimization problem, directly affects the number of SQP iterations needed for the controller to converge to a solution.
 505 A higher number of SQP iterations increases the computational time.

Although this work does not focus on optimizing the computational performance of the formulated controllers to ensure
 real-time feasibility, the impact of the proposed model adaptation on the computational performance of the controllers is
 assessed.

Figure 9 presents the economic profit and computational time for ENMPC_{aug} LiDaR, with the controller limited to a maxi-
 510 mum of three, five, and ten SQP iterations, as reported on the x -axis. The results are normalized with respect to the ENMPC
 LiDaR formulation, which uses a maximum of ten SQP iterations. The y -axis on the left shows the relative economic profit
 in blue, while the y -axis on the right displays the relative mean CPU time per controller iteration in orange. The results for
 ENMPC_{aug} LiDaR are marked with circular markers filled in blue, and the results for ENMPC LiDaR are marked with circular
 markers filled in green.

515 The simulations were performed on a desktop computer with an Intel i7 processor, a 64-bit operating system, and 8 gigabytes
 of RAM.

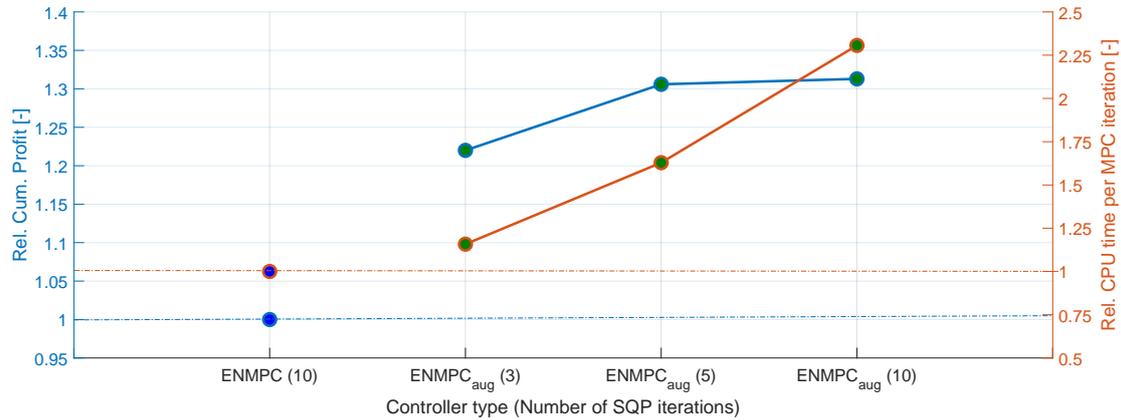


Figure 9. Economic performance and the corresponding CPU computational time requirement for several ENMPC_{aug} LiDaR formulations utilizing an increasing number of SQP iterations. Performance is normalized with respect to the ENMPC LiDaR formulation utilizing ten SQP iterations.

The results clearly show that increasing the number of SQP iterations leads to a higher relative economic profit. For ten SQP iterations, ENMPC_{aug} LiDaR results in a 30% higher economic profit, but it requires more than double the computational effort compared to ENMPC LiDaR. This is because of the additional calculations in the NN part of the model.

520 Interestingly, ENMPC_{aug} LiDaR with only five SQP iterations still achieves a 22% higher economic profit than ENMPC LiDaR with ten SQP iterations, while requiring just 15% more computational effort. This demonstrates that, although the proposed model adaptation increases the CPU time for ENMPC, the improvement in economic performance outweighs the additional computational cost. Moreover, the adapted model requires fewer SQP iterations to generate economically optimal control actions at a smaller actuator usage.

525 This effect, combined with the use of high performance computational platforms and further software optimization for speed, may result in real-time feasible economic controllers.

5 Conclusions

MPC uses an approximate representation of the plant to predict the system evolution over a short future horizon. The internal models used by MPC are typically a reduced representation of the plant. The resulting model mismatch affects the feasibility and the optimality of the closed-loop performance. This study presented an adaptive ENMPC for closed-loop control of wind turbines. The ENMPC directly incorporates cyclic fatigue costs using a novel online rainflow counting approach and is able to generate more accurate model predictions through a data-driven adapted model.

The proposed adaptive ENMPC aims to maximize economic profit, which is here computed as the revenue from power generation minus the cost of fatigue damage at the tower base. Revenue is calculated based on the market tariff and the energy



535 supplied to the grid. The cyclic fatigue damage is estimated using online rainflow counting on adapted model predictions, while also accounting for residual cycles. While cycle counting is the only approach to precisely account for fatigue, it also inherently introduces discontinuities in the MPC optimization problem. To address this problem, the PORFC approach is used to externalize fatigue estimation from the MPC optimization. This method estimates time-varying fatigue-related parameters, allowing for a continuous formulation of cyclic fatigue costs that can be numerically optimized within the MPC.

540 Additionally, the ENMPC optimization is subject to bound constraints on system states and control variables, ensuring stable closed-loop behavior. The ENMPC formulation is further enhanced by integrating state and disturbance estimators. These estimators account for measurement uncertainties and provide accurate initial values for the ENMPC.

The proposed approach utilizes wind turbine operational or synthetic data with machine learning techniques to predict the mismatch in the system states of plant and ROM. A data-driven model is developed to estimate the error in the states across a
545 range of relevant inflow and control conditions. This process results in an adapted model, where the underlying dynamics are represented by a simple physics-based model combined with data-driven correction terms. The adapted ROM is then used as the internal model within the controller.

The performance of the proposed approach is assessed through a case study using an OpenFAST model of the NREL 5 MW reference wind turbine as plant. This model generates measurement data for the study. The plant model has fifteen degrees
550 of freedom and includes thirty-three system states, while the ROM has only three degrees of freedom and consists of eight system states. The simulation data is generated across a wide range of operational conditions. A feed-forward NN is developed to predict the error in plant and ROM states, and is then used in the adapted model dynamics.

The performance of the proposed model adaptation is initially evaluated in an open-loop setup. In this case, the predictions of the adapted model ROM_{aug} and the baseline ROM are compared with the actual behavior of the plant across a range of
555 operational conditions. The kernel density estimates of the prediction error show that the adapted model performs significantly better than the original model for all eight system states. The performance improvement is further quantified in terms of statistics of the prediction error. The results reveal a reduction in both the mean and standard deviation of errors for all system states, with an approximate 20% reduction in the angular velocity error of the rotor.

Additionally, the effect of dataset size on the open-loop performance of the proposed model adaptation is assessed. Results
560 indicate that even a relatively small subset of training data helps reduce the plant-model mismatch. Performance starts to plateau once 85% of the generated training dataset is used, as additional data samples no longer significantly improve the model.

To further quantify the impact of model adaptation on the economic control of wind turbines, the closed-loop performance of the proposed economic MPC formulation is assessed by using ROM_{aug} as the internal model of the controller. Five performance indicators are considered: revenue (due to wind power generation), cost (due to tower fatigue damage in fore-aft and side-side
565 directions), profit (calculated as the difference of revenue and cost), pitch travel (as a proxy for actuator usage), and torque travel (as a proxy for damage of power electronic converters and drivetrain usage). The optimization problem is solved with the Acados framework using an interior-point solver. The MPC sample time is chosen as 100 milliseconds with a prediction horizon of 2 seconds, where multiple SQP iterations are performed per controller step.



The performance of ENMPC_{aug} is compared to the ENMPC formulation that uses ROM as its internal model. Results show
570 that ENMPC_{aug} achieves 9% higher economic profit than ENMPC. This improvement is attributed to more accurate revenue
and cost estimations within ENMPC_{aug}, made possible by better predictions of the system states. Additionally, since the future
predictions in ENMPC_{aug} are closer to the actual evolution of the plant, the controller requires relatively fewer control actions.
This is reflected in a significantly smaller pitch and torque travel compared to the ENMPC formulation.

The benefits of the proposed model adaptation are further assessed across different wind input scenarios. The results show
575 an increased economic profit with improved wind foresight, for both ENMPC_{aug} and ENMPC. Moreover, the model adaptation
leads to higher economic profit – up to 30% in the LiDaR scenario – and reduced pitch and torque travel for all wind input
scenarios.

Although this study did not focus on optimizing the computational performance of the controllers for real-time feasibility,
the impact of model adaptation on computational performance was also evaluated. The results show that increasing the number
580 of SQP iterations leads to higher economic profit but also increases computational expenses. Additionally, for the same number
of SQP iterations, ENMPC_{aug} is more computationally expensive than ENMPC due to the extra computational cost due to NN
evaluations. However, when considering LiDaR wind estimates, ENMPC_{aug} achieves a 22% higher economic profit while
requiring only 15% more computational effort compared to ENMPC. Therefore, the proposed model adaptation, combined
with high-performance computing platforms and additional software optimization (not considered here), could enable real-
585 time feasible economic controllers.

The accuracy of offline data-driven model corrections heavily depends on the quality of its training process, which requires a
comprehensive dataset characterizing the range of operational conditions of a wind turbine. In reality, the wind turbine OEMs
typically have validated high-fidelity turbine models from the design and prototyping phases. These models can be used to
represent the wind turbine behavior and generate the dataset. Moreover, OEMs sometimes also have access to high-resolution
590 data from on-board sensors of operational turbines that can also contribute to the required dataset. A generalized usability of
the adapted model relies on how similar the turbine behavior is from one installation site to another, a problem that was not
investigated here.

Future work should focus on developing a more comprehensive economic objective that accounts for the fatigue damage of
additional turbine components, such as the bearings and drivetrain, while also incorporating a more realistic profit evaluation
595 model. Currently, the profit formulation overlooks the impact of fatigue on component reliability and O&M costs, and only
considers tower fatigue damage. Additionally, the physics-based internal model dynamics can be expanded to better capture
dependencies on other system states. Furthermore, the model can be enhanced to include online tuning of the model parameters,
in addition to the offline augmentation used in this study.

Appendix A: Nomenclature and abbreviations

600	DLC	Design load case
	DOF	Degree of freedom



	ENMPC	Economic nonlinear model predictive control
	ENMPC _{aug}	Economic nonlinear model predictive control having ROM _{aug} as the controller-internal model
	FA	Fore-aft
605	LiDaR	Light detection and ranging
	MHE	Moving horizon estimator
	MPC	Model predictive control
	NFWS	Nacelle filtered wind speed
	NLP	Nonlinear program
610	NN	Neural network
	O&M	Operation and maintenance
	ODE	Ordinary differential equation
	OEM	Original equipment manufacturer
	PDE	Probability density estimate
615	PF	Perfect foresight
	PORFC	Parametric online rainflow counting
	QP	Quadratic program
	RAWS	Rotor averaged wind speed
	REWS	Rotor effective wind speed
620	RMSE	Root mean squared error
	ROM	Reduced order model
	ROM _{aug}	Augmented reduced order model
	SQP	Sequential quadratic program
	SS	Side-side
625	STD	Standard deviation
	ξ	Slack variable
	\bar{v}	Noise variable
630	ω	Rotor speed
	β_b	Blade pitch angle
	β_c	Commanded blade pitch angle
	σ	Stress at tower base
	η	Power conversion efficiency of the drivetrain

635



	F	Set of dynamic equations of a model
	F_{NN}	Function mapping of the NN
	ΔF	Set of dynamic equations of the correction model
640	F_T	Aerodynamic thrust force
	J	Optimization objective
	J_r	Moment of inertia of the rotor
	N_u	Number of intervals in the controller prediction horizon
	P	Electrical power output of the turbine
645	R_m	Ultimate tensile strength of the material
	T_g	Generator torque
	T_{gc}	Commanded generator torque
	T_Q	Aerodynamic torque
	T_{ctrl}	Sample time of the internal model and the controller
650	$T_{horizon}$	Prediction horizon of the controller
	$T_{horizon,est}$	Prediction horizon of the state estimator
	T_{sim}	Sample time of the plant
	V_w	Wind speed
	W	Weight
655		
	a_m	Initial capital cost of the machine
	b	Bias
	c	Cycle
660	d	Disturbance variable
	$d_{T_{FA}}$	Tower-top deflection in fore-aft direction
	$d_{T_{SS}}$	Tower-top deflection in side-side direction
	e	Error in state
	f^{act}	Activation function
665	i	Time instant
	m	Time-varying PORFC parameter: mean
	p	Free model parameter
	t	Time
	Δt	Difference in time between current and next instants
670	u	Control variable
	w	Time-varying PORFC parameter: weight



	x	State variable
	x_{NN}	Input feature of the NN
	y_{NN}	Output feature of the NN
675		
	b_{1-2}	Model parameters for blade dynamics
	f_{1-3}	Model parameters for tower fore-aft dynamics
	g_1	Model parameters for generator dynamics
680	m	Fatigue exponent derived from material properties
	s_{1-4}	Model parameters for tower side-side dynamics
	\square_{aug}	Augmented
685	\square_{est}	Estimation
	\square_{gen}	Generation
	\square_{meas}	Measurement
	\square_{prev}	Previous
	\square_{sim}	Simulation
690		

Code and data availability. A Python script to extract PORFC parameters for a given stress time series can be accessed on Zenodo at <https://doi.org/10.5281/zenodo.15530467> (Anand and Bottasso, 2025). The data for figures 3-9 can also be retrieved in Python pickle format from the same Zenodo repository.

Author contributions. AA and CLB developed the adaptive economic MPC formulation. AA implemented the model correction, the adaptive economic controller, carried out the simulations, and generated results. All authors contributed to the interpretation of the results. CLB supervised the overall research. AA and CLB prepared the manuscript. All authors provided valuable input to this research work through discussions, feedback, and improvement of the manuscript.

Competing interests. The authors declare that they have no conflict of interest, except for CLB who is the Editor-in-Chief of the Wind Energy Science journal.



700 *Financial support.* This work has also been supported by the e-TWINS project (FKZ: 03EI6020), which received funding from the German Federal Ministry for Economic Affairs and Climate Action (BMWK).

Acknowledgements. The authors would like to acknowledge Stefan Loew for his valuable inputs to the overall work, and Filippo Campagnolo for his support in the implementation of different wind inputs.



References

- 705 Anand, A. and Bottasso, C. L.: Adaptive economic wind turbine control [source code and dataset], Zenodo, <https://doi.org/10.5281/zenodo.15530467>, 2025.
- Anand, A., Loew, S., and Bottasso, C. L.: Economic control of hybrid energy systems composed of wind turbine and battery, pp. 2565–2572, <https://doi.org/10.23919/ECC54610.2021.9654911>, 2021.
- Anand, A., Loew, S., and Bottasso, C. L.: Economic nonlinear model predictive control of fatigue for a hybrid wind-battery generation
710 system, *Journal of Physics: Conference Series*, 2265, 032 106, <https://doi.org/10.1088/1742-6596/2265/3/032106>, 2022.
- 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.
- Bonnie Jonkman, Rafael M Mudafort, Andy Platt, E. Branlard, Mike Sprague, jjonkman, HaymanConsulting, Ganesh Vijayakumar, Marshall Buhl, Hannah Ross, Pietro Bortolotti, Marco Masciola, Shreyas Ananthan, Michael J. Schmidt, Jon Rood, rdamiani, nrmendoza,
715 sinolonghai, Matt Hall, ashesh, kshaler, Kurt Bendl, pschuenemann, psakievich, ewquon, mattphillips, Nobuhiro KUSUNO, alvarogonzalezsalcedo, Tony Martinez, and rcorniglian: OpenFAST/openfast: OpenFAST v3.1.0, <https://doi.org/10.5281/zenodo.6324288>, 2022.
- Bottasso, C. L. and Croce, A.: Cascading Kalman observers of structural flexible and wind states for wind turbine control: Scientific Report DIA-SR 09-02, https://www.researchgate.net/publication/228941698_Cascading_Kalman_observers_of_structural_flexible_and_wind_states_for_wind_turbine_control, 2009.
- 720 Bottasso, C. L., Chang, C.-S., Croce, A., Leonello, D., and Riviello, L.: Adaptive planning and tracking of trajectories for the simulation of maneuvers with multibody models, *Computer Methods in Applied Mechanics and Engineering*, 195, 7052–7072, <https://doi.org/10.1016/j.cma.2005.03.011>, 2006.
- Canet, H., Loew, S., and Bottasso, C. L.: What are the benefits of lidar-assisted control in the design of a wind turbine?, *Wind Energy Science*, 6, 1325–1340, <https://doi.org/10.5194/wes-6-1325-2021>, 2021.
- 725 Collet, D., Alamir, M., Di Domenico, D., and Sabiron, G.: Data-driven fatigue-oriented MPC applied to wind turbines Individual Pitch Control, *Renewable Energy*, 170, 1008–1019, <https://doi.org/10.1016/j.renene.2021.02.052>, 2021.
- 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.
- Huang, R., Biegler, L. T., and Patwardhan, S. C.: Fast Offset-Free Nonlinear Model Predictive Control Based on Moving Horizon Estimation,
730 *Industrial & Engineering Chemistry Research*, 49, 7882–7890, <https://doi.org/10.1021/ie901945y>, 2010.
- Jonkman, B. J.: Turbsim User’s Guide: Version 1.50, <https://doi.org/10.2172/965520>, 2009.
- Jonkman, J., Butterfield, S., Musial, W., and Scott, G.: Definition of a 5-MW Reference Wind Turbine for Offshore System Development, <https://doi.org/10.2172/947422>, 2009.
- Loew, S. and Bottasso, C. L.: Lidar-assisted model predictive control of wind turbine fatigue via online rainflow counting considering stress
735 history, *Wind Energy Science*, 7, 1605–1625, <https://doi.org/10.5194/wes-7-1605-2022>, 2022.
- Loew, S., Obradovic, D., and Bottasso, C. L.: Direct Online Rainflow-counting and Indirect Fatigue Penalization Methods for Model Predictive Control, pp. 3371–3376, <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, 2020a.



- 740 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, 022 041, 2020b.
- Loew, S., Anand, A., and Szabo, A.: Economic model predictive control of Li-ion battery cyclic aging via online rainflow-analysis, *Energy Storage*, 3, <https://doi.org/10.1002/est2.228>, 2021.
- Loew, S., Obradovic, D., and Bottasso, C. L.: Economic nonlinear model predictive control of fatigue—Formulation and application to wind turbine control, *Optimal Control Applications and Methods*, 44, 647–676, <https://doi.org/10.1002/oca.2870>, 2023.
- 745 Nafis Faizi and Yasir Alvi: Chapter 6 - Correlation**For datasets, please refer to companion site: <https://www.elsevier.com/books-and-journals/book-companion/9780443185502>, in: *Biostatistics Manual for Health Research*, edited by Nafis Faizi and Yasir Alvi, pp. 109–126, Academic Press, ISBN 978-0-443-18550-2, <https://doi.org/10.1016/B978-0-443-18550-2.00002-5>, 2023.
- Rawlings, J. B., Angeli, D., and Bates, C. N.: *Fundamentals of economic model predictive control*, pp. 3851–3861, 750 <https://doi.org/10.1109/CDC.2012.6425822>, 2012.
- 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., ISBN 9780975937730, 2017.
- Ritter, B.: *Nonlinear State Estimation and Noise Adaptive Kalman Filter Design for Wind Turbines*, <https://doi.org/10.25534/tuprints-00011785>, 2020.
- 755 S. Gros: An economic NMPC formulation for wind turbine control, in: *52nd IEEE Conference on Decision and Control*, pp. 1001–1006, <https://doi.org/10.1109/CDC.2013.6760013>, 2013.
- Schreiber, J., Bottasso, C. L., Salbert, B., and Campagnolo, F.: Improving wind farm flow models by learning from operational data, *Wind Energy Science*, 5, 647–673, <https://doi.org/10.5194/wes-5-647-2020>, 2020.
- Soloperto, R., Muller, M. A., and Allgower, F.: Guaranteed closed-loop learning in Model Predictive Control, *IEEE Transactions on Automatic Control*, p. 1, <https://doi.org/10.1109/TAC.2022.3172453>, 2022.
- 760 Soltani, M. N., Knudsen, T., Svenstrup, M., Wisniewski, R., Brath, P., Ortega, R., and Johnson, K.: Estimation of Rotor Effective Wind Speed: A Comparison, *IEEE Transactions on Control Systems Technology*, 21, 1155–1167, <https://doi.org/10.1109/TCST.2013.2260751>, 2013.
- Stehly, T., Beiter, P., and Duffy, P.: 2019 Cost of Wind Energy Review, <https://doi.org/10.2172/1756710>, 2020.
- 765 TotalControl: D2.1 Cost Model for fatigue degradation and O&M - Advanced integrated supervisory and wind turbine control for optimal operation of large Wind Power Plants: [TotalControl], <https://doi.org/10.3030/727680>, 2022.
- Verschueren, R., Frison, G., Kouzoupis, D., van Duijkeren, N., Zanelli, A., Novoselnik, B., Frey, J., Albin, T., Quirynen, R., and Diehl, M.: *acados: a modular open-source framework for fast embedded optimal control*, arXiv preprint, 2019.