Experimental Testing of a Preview-enabled Model Predictive Controller for Blade Pitch Control of Wind Turbines

Michael Sinner*, Vlaho Petrović, Apostolos Langidis, Lars Neuhaus, Michael Hölling, Martin Kühn, and Lucy Y. Pao

***Abstract*—Model predictive control (MPC) is a control method that involves determining the input to a dynamical system as the solution to an optimization problem that is solved online. In the wind turbine research literature, MPC has received considerable attention for its ability to handle both actuator constraints and preview disturbance information about the oncoming wind, which can be provided by a lidar scanner. However, while many studies simulate the wind turbine response under MPC, very few physical tests have been carried out, likely due in part to the difficulties associated with solving the MPC problem in real time. In this work, we implement MPC on an experimental, scaled wind turbine operating in a wind tunnel testbed, using an active grid to create reproducible wind sequences and a hot-wire anemometer to generate upstream wind measurements. To our knowledge, this work presents the first physical test of MPC for blade pitch control of a scaled wind turbine. We compare two MPC strategies: one including preview disturbance information and one without. Our results provide further evidence that feedforward control can improve wind turbine performance in transition and above-rated conditions without increasing actuation requirements, which we hope will encourage industry experimentation and uptake of feedforward control methods. We also provide a high-level analysis and interpretation of the computational performance of the chosen approach. This work builds upon the results of an earlier study, which considered unconstrained optimal blade pitch control.**

***Index Terms*—Feedforward control, real-time model predictive control, wind turbines, wind energy.**

I. Introduction

RESEARCH into model predictive control of wind turbines began in the late 2000s [1]–[3] and has been a popular method of utilizing feedforward disturbance measurements since the early 2010s [4]–[8], after Harris et al. demonstrated in 2005 the possibility of generating measurements of the incoming wind field using a lidar scanner [9]. Many competing model predictive control (MPC) formulations are to be found in the wind turbine literature, including so-called 'linear' MPC [6], [10], [11], nonlinear MPC [11]–[13], economic MPC [14], [15], and robust variations [16], [17] (not all of which are mutually exclusive). Specific to the wind turbine control problem, MPC has mostly focused on rotor speed regulation [18], [19] and load minimization [13], [20], [21] in above-rated winds (for a brief description of wind turbine control operation, see Section III-A; for a more thorough treatment, see Pao & Johnson [22]). While some studies consider the full operating regime [7], [12], [23]–[25], the general conclusion is that MPC, and feedforward control in general [26], appears to be most beneficial in above-rated conditions. An exception is the recent work of Mulders et al., who successfully employ MPC to avoid tower resonances in below-rated operation [27].

While a substantial body of work exists developing preview-enabled MPC methods for wind turbines (of which we have mentioned only a few—many other studies exist in the literature [28]), practical implementations are scarce. On the other hand, there have been reports of physical testing of other feedforward control paradigms, both on full-scale turbines [29]–[33] and model turbine testbeds [19], [34]. A possible explanation for this is that MPC demands a relatively high computational load, since, in general, an optimization problem must be solved online at each controller time step. To address this issue, Gros et al. [15], [35] and Bottasso et al. [24] present work on real-time MPC for wind turbines. Running simulations on desktop computers, Gros et al. achieve a fast enough computation to run a controller at 5 Hz [35], and more recently, Gros & Schild achieve computations that appear to be fast enough for 100 Hz updates (although the controller implemented uses a 10 Hz sampling frequency) [15], while Bottasso et al. achieve speeds compatible with a sampling rate of over 50 Hz [24]. However, we could not find evidence that the authors had implemented their proposed methodology on a physical testbed and, as we discuss in Sections IV & VI, other considerations must be taken care of in a true implementation for closed-loop control. We note that the studies of Gros et al. and Bottasso et al. are not directly comparable as their MPC formulations differ substantially.

To our knowledge, only one physical test of MPC for wind turbines has been carried out to date. Verwaal et al. implemented MPC for torque control on a 1.5 m rotor diameter scaled wind turbine model in a wind tunnel test facility at the Delft University of Technology, utilizing preview wind measurements generated by a pitot tube located upstream of the turbine [19]. Their study compared MPC and a model-inverse feedforward technique to a baseline feedback controller for above-rated turbine operation, and found that, for

*Corresponding author. M. Sinner and L. Y. Pao are with the Department of Electrical, Computer, and Energy Engineering, University of Colorado Boulder, Boulder, CO 80309 USA (e-mail: michael.sinner@colorado.edu, pao@colorado.edu).

V. Petrović, A. Langidis, L. Neuhaus, M. Hölling, and M. Kühn are with ForWind – Center for Wind Energy Research, Institute of Physics, University of Oldenburg, Oldenburg, Germany (e-mail: vlaho.petrovic@uol.de, apostolos.langidis@uol.de, lars.neuhaus@uol.de, michael.hoelling@uol.de, martin.kuehn@uol.de)

deterministic winds, significant improvements could be made in rotor speed regulation. However, without active blade pitch control or automated wind profile generation, the authors were limited to slightly unusual torque-controlled above-rated operation and relatively slowly-varying winds that did not activate actuator constraints. The authors mentioned these shortcomings and pointed to pitch control as a next step in their recommendations for future work [19]. On the other hand, Verwaal et al. presented the first study to successfully test MPC for wind turbines, and we will refer back to their work at several points throughout this paper. In the present work, we address the other side of the picture: MPC for blade pitch control in above-rated and transition operation using faster gusts and turbulent winds, where actuator constraints are activated frequently.

In a previous study, we considered a finite-horizon optimization problem utilizing preview disturbance information for wind turbine blade pitch control [34]. The formulation was similar to the MPC formulation presented here, with the critical difference that physical inequality constraints were not taken into account: this resulted in a linear feedback/feedforward control law that can be considered an extension to the linear-quadratic regulator, which can be computed largely offline. We tested the controller on a scaled wind turbine model [36] operating in a wind tunnel with inflow generated by an active grid [37], as we do in the present study. Results showed that the addition of preview disturbance measurements could significantly improve rotor speed regulation performance; however, testing was limited to relatively slow gust events in strictly above-rated conditions. This was due, in part, to the lack of constraint handling in the linear feedback/feedforward law precluding a good way of saturating the blade pitch angle at its minimum position. The work we presented in Sinner et al. [34] could be considered as a stepping-stone towards the full MPC implementation presented here. As such, we will also refer back to that work frequently in this paper.

The remainder of this paper is laid out as follows. Section II describes the test turbine and simplifications made to arrive at the linear model used for control purposes. Section III begins with a background on wind turbine control for unfamiliar readers, and then presents the proposed MPC used for this work. Section IV describes the physical testbed used for controller evaluation, and Section V presents the test results. Section VI discusses the computational performance of the MPC implementation. Concluding remarks are given in Section VII.

II. Modeling

Testing on a scaled turbine allows us to verify real-time behavior in a controlled environment, without the significant expense of a utility-scale turbine. The ForWind Model Wind Turbine Oldenburg 1.8 (MoWiTO 1.8, which we will refer to as MoWiTO) is a scaled wind turbine model used for testing at ForWind – Center for Wind Energy Research at the University of Oldenburg [36]. MoWiTO is a 1.8 m rotor-diameter turbine designed as an aerodynamically scaled model of the NREL 5MW reference turbine [38], although the tower and blades are considerably stiffer than would be representative of the NREL 5MW turbine. The blade chord Reynolds number is on the order of 100 times lower than that of the NREL 5MW reference turbine, so the blades have been designed to attain a lift distribution similar to the NREL 5MW turbine under lower Reynolds numbers. See Berger et al. [36] for details. MoWiTO is fully actuated with generator torque control and individual blade pitch control (Figure 1). For safety purposes while testing experimental controllers, the turbine is derated from its designed rated power of 363 W [36], and for this study was used with a modified rated power of 140 W and rated rotor speed of 480 rpm. We use only collective blade pitch control (i.e., the pitch command sent to each blade is identical) in this work.

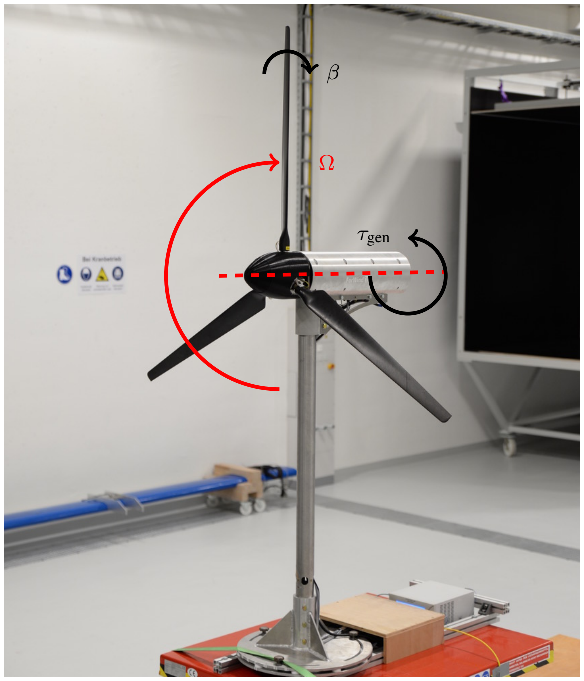


Fig. 1. MoWiTO test turbine. Ω is the rotor rotational velocity, τ_{gen} is the electrical torque supplied by the generator, and β is the pitch angle of the blades.

A medium-fidelity simulation model of MoWiTO is used for controller verification and tuning. The model is implemented in FAST v8 [39], [40], a wind turbine modeling package that links the aerodynamic, elastic, mechanical, and electrical behaviors of the turbine and allows time-domain simulation. FAST allows for turbulent spatially- and temporally-varying inflow winds [41], as well as simpler wind cases. We use FAST embedded in a Simulink model for simulating closed-loop control as well as low-order model fitting (described in Section II-A).

We implement model predictive control using a linear discrete-time state-space model of the plant

$$x(k+1) = Ax(k) + Bu(k) + B_d d(k) \tag{1a}$$
$$y(k) = Cx(k)\,, \tag{1b}$$

which explicitly includes disturbances $d(k) \in \mathbb{R}^{m_d}$ and the disturbance input matrix $B_d \in \mathbb{R}^{n \times m_d}$. Vector quantities

$x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$, and $y(k) \in \mathbb{R}^p$; and matrix quantities $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and $C \in \mathbb{R}^{p\times n}$ have their usual meanings from control theory for discrete-time systems, where (k) denotes the kth time step of the discrete-time system. The inclusion of the disturbance input d allows us to design feedforward control laws that explicitly take the effect of incoming disturbances into account. For simplicity in this demonstration, we have chosen to use a linear time-invariant model (1) for MPC that is valid only for a region around a nominal operating point. Improved results may be seen using a linear parameter-varying model [42], at the expense of a slightly more complicated MPC implementation and model identification procedure.

A. Physics-based Model

Many works (e.g. Pao & Johnson [22]) model the wind turbine as a nonlinear first-order system (see Figure 1)

$$J\dot{\Omega} = \tau_{\text{aero}}(v, \Omega, \beta) - \tau_{\text{gen}}. \tag{2}$$

J is the rotational inertia of the rotor and drivetrain, Ω is the rotor rotational velocity, and v is the wind speed. The torques applied to the rotational system, τ_{aero} and τ_{gen}, refer to the aerodynamic torque produced by the blades and the electrical torque produced by the generator, respectively. For the purpose of this work, we are concerned with regulating the rotor speed using blade pitch β considering the effects of disturbances in the wind speed v.

To linearize the system dynamics (2), a standard approach is to take a first-order Taylor series expansion about an operating point and arrive at the linearized model

$$\begin{aligned}\dot{\Omega} \approx \frac{1}{J}\frac{\partial \tau_{\text{aero}}}{\partial \Omega}\bigg|_{\text{nom}} \delta\Omega + \frac{1}{J}\frac{\partial \tau_{\text{aero}}}{\partial \beta}\bigg|_{\text{nom}} \delta\beta \\ + \frac{1}{J}\frac{\partial \tau_{\text{aero}}}{\partial v}\bigg|_{\text{nom}} \delta v + \frac{1}{J}\delta\tau_{\text{gen}}\end{aligned} \tag{3}$$

where $|_{\text{nom}}$ indicates that the derivatives are evaluated at some nominal equilibrium operating condition (for which $\dot{\Omega}_{\text{nom}} = 0$) and $\delta w \overset{\text{def}}{=} w - w_{\text{nom}}$ for some signal w. To find an appropriate trim condition we use FAST simulations. From previous test data from MoWiTO [34], setting the blade pitch angle at $\beta = 4.53°$ and generator torque at rated ($\tau_{\text{gen,rated}} = 2.8$ Nm) when the wind speed is $v = 7.2$ m/s should produce the rated rotor speed. However, the current FAST model for MoWiTO does not account for frictional losses, so simulating with these inputs resulted in a rotor speed considerably higher than rated. To overcome this issue and provide a linearized model with accurate aerodynamic properties, we increased the generator torque in simulation until the rated rotor speed was achieved. This procedure differs from our previous work [34]. The final trim condition used for linearization is listed in Table I, where the $_{\text{nom}}$ subscript denotes nominal operation.

TABLE I
Above-rated Trim Condition in Simulation

Wind speed v_{nom} [m/s]	7.2
Rotor speed $\Omega_{\text{rpm,nom}}$ [rpm]	480 (rated)
Generator torque $\tau_{\text{gen,nom}}$ [Nm]	4.027
Blade pitch β_{nom} [°]	4.53

Instead of evaluating the derivatives in the linearized model (3) directly, we use a simple system identification procedure to fit the linear first-order model to the response from the nonlinear (higher-order) FAST model. Beginning at the trim condition, we provided the FAST model with both a step blade pitch input of 5° and a step wind input of 0.5 m/s (Figure 2). The degrees of freedom included in simulation were the turbine rotational degree of freedom and the first flapwise mode for each blade. The wind input was generated as uniform across the rotor disk. Since our MPC is designed for pitch control only, the transfer function from generator torque to rotor speed is not needed. The resulting data is enough to fit the two first-order transfer function models: one from blade pitch to rotor speed and one from wind speed to rotor speed. The mean time constant of these models can be chosen to represent the time constant of the system, and the system can be written in the continuous-frequency domain as $Y(s) = G(s)U(s)$, where $Y(s)$ is the frequency domain rotor speed output, $U(s)$ stacks the blade pitch and wind disturbance inputs in the frequency domain, and $G(s) = \begin{bmatrix} \frac{cb}{s-a} & \frac{cb_d}{s-a} \end{bmatrix}$ is a transfer function matrix. According to our model (3), we have that

$$a \overset{\text{def}}{=} \frac{1}{J}\frac{\partial \tau_{\text{aero}}}{\partial \Omega}\bigg|_{\text{nom}}, \quad b \overset{\text{def}}{=} \frac{1}{J}\frac{\partial \tau_{\text{aero}}}{\partial \beta}\bigg|_{\text{nom}}, \quad b_d \overset{\text{def}}{=} \frac{1}{J}\frac{\partial \tau_{\text{aero}}}{\partial v}\bigg|_{\text{nom}},$$

and $c \overset{\text{def}}{=} {}^{60}/_{2\pi}$, so that we can give our system the continuous-time state-space realization

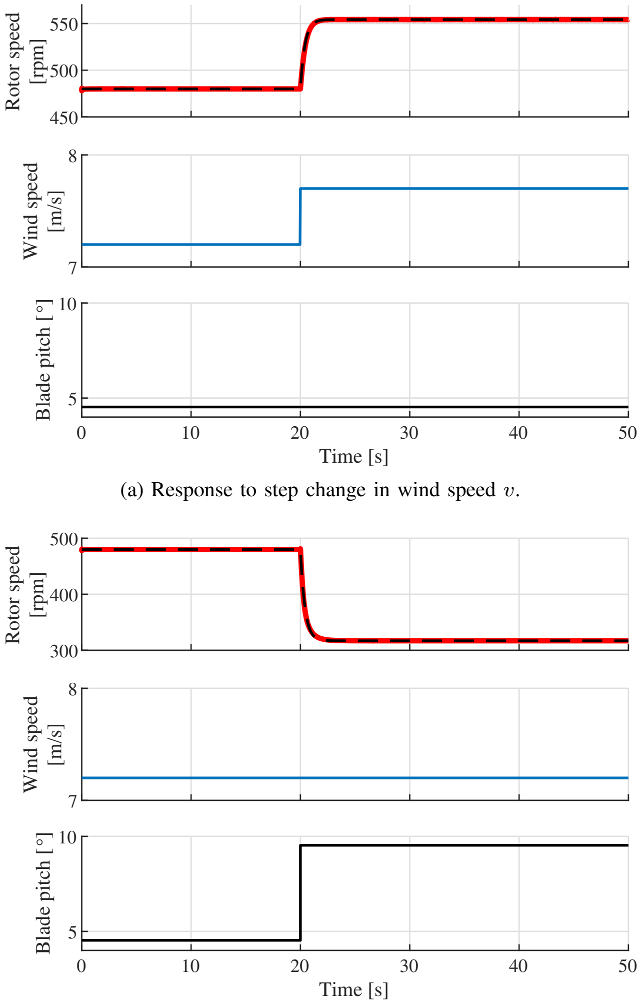
$$\dot{x} = ax + bu + b_d d \tag{4a}$$

$$y = cx\,. \tag{4b}$$

B. Mathematical Extensions

Although physically we only model the rotor rotational degree of freedom, we augment our model (4) to include the integral of the rotor speed error so that we can perform integral control action [43, pp. 549–550]. Next, we discretize the model via zero-order hold at the controller sampling rate $f_s = 100$ Hz. As a final step, we add a state that accumulates the system input and reformulate the blade pitch input as a difference in pitch angle between two time steps. This allows us to penalize the pitch rate in the MPC formulation, which is of concern in terms of actuator usage, as opposed to the absolute pitch position, which is not of concern except when considering constraint satisfaction (Section III-B2). The applied pitch angle β is recovered as $\beta(k) = \beta(k-1) + u(k)$.

We let Δ represent a change in a quantity from the previous time step to the current time (whereas δ denotes a deviation from nominal operation). For instance, for some physical signal $w(k)$, $\Delta w(k) \overset{\text{def}}{=} w(k) - w(k-1)$. Note that, because the nominal operating point w_{nom} is fixed, $\Delta w(k) = \delta w(k) - \delta w(k-1)$. We can now express our model



(a) Response to step change in wind speed v.

(b) Response to step blade pitch input β.

Fig. 2. Rotor speed responses (top plots) to step inputs (lower plots). The red line represents the full nonlinear FAST model, while the black dashed line represents the linearized model fit.

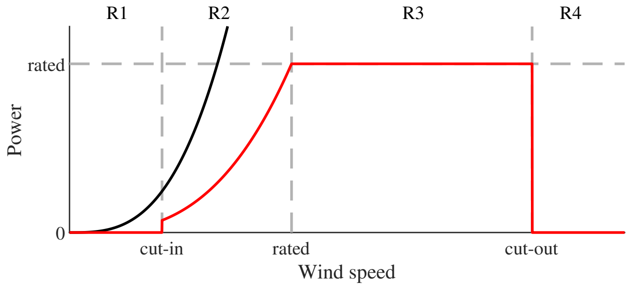


Fig. 3. Power curve of a modern wind turbine as a function of wind speed (red curve), with the power in the wind marked in black. Regions of operation are marked above (R1–R4).

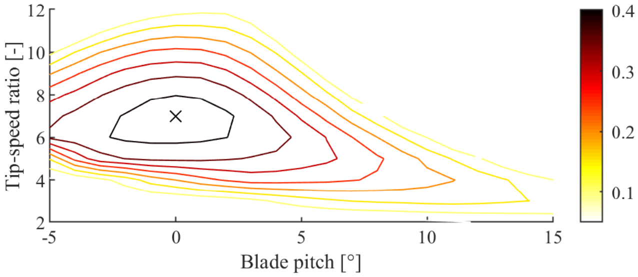


Fig. 4. Power coefficient (C_P) surface for MoWiTO. The black × denotes the maximum point, which is used for Region 2 operation.

in the form of (1), with the state, control input, disturbance input, and output

$$x(k) = \begin{bmatrix} \delta\Omega(k) \\ \int \delta\Omega(k) \\ \delta\beta(k-1) \end{bmatrix}, \; u(k) = \Delta\beta(k),$$
$$d(k) = \delta v(k), \text{ and } y(k) = \begin{bmatrix} \delta\Omega_{\text{rpm}}(k) \\ \int \delta\Omega_{\text{rpm}}(k) \end{bmatrix},$$

respectively. Ω represents the rotor speed in radians per second, whereas Ω_{rpm} represents the rotor speed in revolutions per minute (as measured by an encoder). By $\int \delta\Omega(k)$ (resp. $\int \delta\Omega_{\text{rpm}}(k)$), we mean $\int_0^{t(k)} \delta\Omega(t)dt$ (resp. $\int_0^{t(k)} \delta\Omega_{\text{rpm}}(t)dt$), the integral of the rotor speed deviation from nominal in rad/s (resp. rpm) from the beginning of controller time until the current time $t(k) = {}^{k}/_{f_s}$. Note that $x(k)$ has $\delta\beta(k-1)$ as the third element. Our final discrete-time model matrices for MoWiTO are

$$A = \begin{bmatrix} 0.98 & 0 & -4.6 \\ 0.0099 & 1 & -0.023 \\ 0 & 0 & 1 \end{bmatrix}, \; B = \begin{bmatrix} -4.6 \\ -0.023 \\ 1 \end{bmatrix},$$
$$B_d = \begin{bmatrix} 0.37 \\ 0.0019 \\ 0 \end{bmatrix}, \text{ and } C = \begin{bmatrix} 9.6 & 0 & 0 \\ 0 & 9.6 & 0 \end{bmatrix}.$$

III. Control

A. Background on Wind Turbine Control

The operation of industry-standard horizontal-axis wind turbines is generally split into four regions according to the mean wind speed. In Region 1, the wind speed is too low for economically-viable energy generation, and the turbine is parked. Once the cut-in wind speed is reached, the turbine enters Region 2 (also known as 'below-rated' or 'partial load' operation), in which the control goal is to extract as much power as possible from the wind. As the wind speed increases, the turbine reaches a condition where it is generating its rated power. From this point on, the goal is to regulate the turbine at its rated rotor speed, while maintaining rated torque, to continue to produce rated power. This mitigates structural loading on the turbine components and electrical loading on the turbine generator while still producing power, and is referred to as Region 3, 'above-rated' or 'full load' operation. Finally, as wind speed increases to the cut-out wind speed, the turbine can no longer operate safely and is shut down (Region 4). See Figure 3 for a schematic.

The majority of research focuses on the power-generating Regions, i.e. 2 and 3. Two main control inputs are used

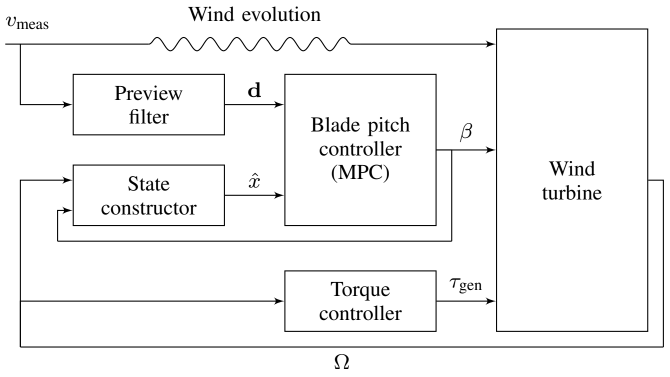


Fig. 5. Block diagram of wind turbine control system used in this study.

for actuation: the generator electrical torque τ_{gen} and the collective blade pitch angle β (see Figure 1). In Region 2, the blades are pitched at a fixed position that maximizes the power coefficient C_P, which represents the ratio between power produced and the power available in the wind. The generator torque is then varied to vary the rotor speed and maintain the C_P-maximizing tip-speed ratio λ (Figure 4). In Region 3, the generator torque is usually held constant at rated torque while the blades are actively pitched to maintain a constant aerodynamic torque τ_{aero} and regulate the rotor speed to rated speed. This combination (rated torque and rated speed) produces the rated turbine power. In addition to the Regions mentioned above, there is often a transition between Regions 2 and 3 (sometimes referred to as Region 2.5) as the turbine switches from torque control to blade pitch control.

B. Model Predictive Controller

We present the general form for the linear-quadratic ('linear') MPC problem first, and provide problem specifics in Sections III-B1 & III-B2. The linear MPC problem is a quadratic program

$$\underset{\{x(i)\},\{u(i)\}}{\text{minimize}} \sum_{i=0}^{N-1} \left\{ x(i)^{\mathsf{T}} Q x(i) + u(i)^{\mathsf{T}} R u(i) \right\} + x(N)^{\mathsf{T}} P x(N) \tag{5a}$$

$$\text{subject to } x(0) = \hat{x}(k) \tag{5b}$$

$$x(i+1) = Ax(i) + Bu(i) + B_d d(k+i), \quad i = 0, ..., N-1 \tag{5c}$$

$$Jx(i) + Eu(i) \leq \ell, \quad i = 0, ..., N-1 \tag{5d}$$

where $x(i)$ and $u(i)$ are the predicted state and control at future time step i; $Q \in \mathbb{R}^{n\times n}$, $R \in \mathbb{R}^{m\times m}$, and $P \in \mathbb{R}^{n\times n}$ are the cost function matrices (Section III-B1); and $N \in \mathbb{N}$ is the prediction (and preview) horizon length. $J \in \mathbb{R}^{r\times n}$, $E \in \mathbb{R}^{r\times m}$, and $\ell \in \mathbb{R}^{r}$ represent the system constraints (Section III-B2), where r is the number of scalar affine constraints for each time step. The inclusion of the disturbance effect $B_d d(k)$ is non-standard, since for most systems, the disturbance is unknown. Only the first step $u(0)$ of the optimal sequence $\{u(i)\}_{i=0,1,\dots,N-1}$ is implemented—afterwards, a new initial state $\hat{x}$ becomes available and the problem is solved again.

Physically, the cost function (5a) represents the objective to be minimized; constraint (5b) sets the initial condition for the MPC problem to the current state (here, we have denoted real time steps by k and predicted time steps by i—various notations are used for this distinction in the MPC literature); constraint (5c) enforces the behavior of the linear model (1); and the inequality constraint (5d) represents physical system constraints. In general, the current system state $\hat{x}(k)$ would need to be estimated online based on the feedback measurements y, control input u, and disturbance d if available, by a state observer. In this study, the state vector x chosen (see Section II-B) is simple enough that we can reconstruct the current state $\hat{x}(k)$ from direct measurements of the rotor speed Ω and blade pitch angle β. We denote this procedure as the 'State constructor' in Figure 5, which would be replaced by an observer in a more complex case.

The MPC problem (5) is commonly written in matrix-vector form

$$\underset{\mathbf{x},\mathbf{u}}{\text{minimize}} \; \mathbf{x}^{\mathsf{T}}\mathbf{Q}\mathbf{x} + \mathbf{u}^{\mathsf{T}}\mathbf{R}\mathbf{u} \tag{6a}$$

$$\text{subject to } \mathbf{x} = \mathbf{A}\hat{x}(k) + \mathbf{B}\mathbf{u} + \mathbf{B}_d\mathbf{d}(k) \tag{6b}$$

$$\mathbf{J}\mathbf{x} + \mathbf{E}\mathbf{u} \leq \boldsymbol{\ell} \tag{6c}$$

(see Appendix A for matrix definitions) and then condensed by substituting the model (6b) into the cost function (6a) and constraint (6c) to arrive at

$$\underset{\mathbf{u}}{\text{minimize}} \; \mathbf{u}^{\mathsf{T}}\mathbf{H}\mathbf{u} + 2\mathbf{h}^{\mathsf{T}}\mathbf{u} \tag{7a}$$

$$\text{subject to } \mathbf{G}\mathbf{u} \leq \mathbf{g} \tag{7b}$$

where

$$\mathbf{H} = \mathbf{B}^{\mathsf{T}}\mathbf{Q}\mathbf{B} + \mathbf{R}\,, \mathbf{h} = \mathbf{B}^{\mathsf{T}}\mathbf{Q}\begin{bmatrix}\mathbf{A} & \mathbf{B}_d\end{bmatrix}\begin{bmatrix}\hat{x}(k)\\ \mathbf{d}(k)\end{bmatrix},$$

$$\mathbf{G} = \mathbf{J}\mathbf{B} + \mathbf{E}, \text{ and } \mathbf{g} = \boldsymbol{\ell} - \mathbf{J}\begin{bmatrix}\mathbf{A} & \mathbf{B}_d\end{bmatrix}\begin{bmatrix}\hat{x}(k)\\ \mathbf{d}(k)\end{bmatrix}.$$

This is the form of the problem that is solved at run-time.

1) Cost Function: The cost function (5a) that we use for MPC is identical to that used in our previous work looking at unconstrained optimal control [34]. We again use $Q = C^{\mathsf{T}}LC$ in order to penalize deviations in the plant output $y(k)$, where

$$L = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix},$$

and set $R = 1 \times 10^5$. It is likely that improved performance could be achieved using a more carefully-tuned cost function, but we found that these weights were adequate for demonstrative purposes. Following MPC literature on extending the quadratic cost function to an infinite horizon [44], the terminal weight P is chosen as the solution to the discrete-time algebraic Riccati equation

$$P = A^{\mathsf{T}}PA + Q - A^{\mathsf{T}}PB(B^{\mathsf{T}}PB + R)^{-1}B^{\mathsf{T}}PA\,. \tag{8}$$

Note that the inclusion of the terminal cost $x(N)^{\mathsf{T}}Px(N)$ is only equivalent to extending the prediction horizon to infinity if both $d(k+i) = 0$ and the constraint

(5d) is satisfied under the linear-quadratic regulator law $u(i) = -\left(B^{\mathsf{T}}PB + R\right)^{-1} B^{\mathsf{T}}PAx(i)$ for all $i \geq N$. In our case, these conditions do not hold; however, we found that using a terminal cost relating to the unconstrained (and undisturbed) system is still a useful approximation.

2) Constraints: The preliminary study for this work [34] did not consider constraints in the optimal control problem; that is, we solved problem (5) without considering the constraints (5d). However, to better represent the physical limitations of the turbine and find controls that are optimal given those limitations, we now add constraints into the optimization. This is the main extension of this controller over the previous, as $u(k)$ is now determined by a nonlinear control law that involves solving the MPC problem online, rather than the linear control law presented in Sinner et al. [34], which could be found offline.

Two major constraints are considered: a blade pitch rate constraint and a blade pitch angle constraint. Since the control input u is defined as the blade pitch increment (see Section II-B), the rate constraint is simply the box constraint

$$-\dot{\beta}_{\text{max}}T_s \leq u(i) \leq \dot{\beta}_{\text{max}}T_s, \quad i = 0, ..., N-1 \tag{9}$$

where $T_s = 0.01$ s is the controller sampling time. On the other hand, the incremental form for the input means that the previous blade pitch angle $\delta\beta(k-1)$ is included when enforcing the blade pitch angle limit, i.e.

$$u(k) + \delta\beta(k-1) \geq \beta_{\text{min}} - \beta_{\text{nom}}\,. \tag{10}$$

Combining constraints (9) and (10), we have that

$$J = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \; E = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}, \text{ and } \ell = \begin{bmatrix} \dot{\beta}_{\text{max}}T_s \\ \dot{\beta}_{\text{max}}T_s \\ \beta_{\text{nom}} - \beta_{\text{min}} \end{bmatrix}.$$

3) Configurations: We consider both a feedback only (FB only) and feedback/feedforward (FB/FF) configuration for MPC. This represents the difference between having preview disturbance measurements available for feedforward control and having only standard feedback channels. In the FB/FF configuration, the vector $\mathbf{d}$ of future disturbances is assumed to be known over the prediction horizon N. For utility-scale wind turbines, this measurement can be provided by a lidar [9], but for the present scaled testing, we instead use a hot-wire anemometer to generate preview measurements. See Section IV-D for details. The problem is unchanged for the FB only case, except that we set $d(k) \equiv 0 \,\forall\, k$ (or $\mathbf{d}(k) \equiv \mathbf{0}$), which is equivalent to saying that the disturbance is completely unknown. This does not change the problem size, since the decision variable $\mathbf{u} \in \mathbb{R}^{Nm}$ in the condensed problem (7) has the same dimension whether or not $\mathbf{d}$ is set to zero. The full list of parameters chosen for the MPC problem are provided in Table II.

C. Torque Control

The model predictive controller described controls only the blade pitch angle β. To handle Region 2 (below-rated) operation and transition, we also implement a generator torque controller (see Figure 5).

TABLE II
MODEL PREDICTIVE CONTROLLER PARAMETERS

Prediction horizon length N	10
Controller sampling rate f_s [Hz]	100
Output cost matrix L	I_2
Input cost matrix R	1×10^5
Blade pitch angle limit β_{min} [°]	0
Blade pitch rate limit $\dot{\beta}_{\text{max}}$ [°/s]	86
Allowable solve time T_{solve} [ms]	3

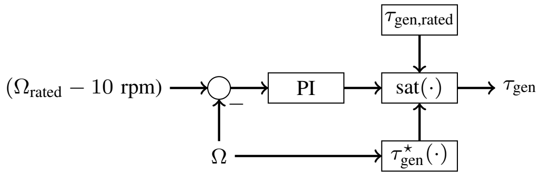

$\tau_{\text{gen,rated}}$
$(\Omega_{\text{rated}} - 10$ rpm$)$
$-$
PI
sat$(\cdot)$
τ_{gen}
Ω
$\tau^{\star}_{\text{gen}}(\cdot)$

Fig. 6. Simplified diagram of the torque controller used. The saturation has a lower limit of $\tau^{\star}_{\text{gen}}(\Omega)$, the optimal power-producing torque for a given rotor speed, and an upper limit of the rated torque $\tau_{\text{gen,rated}}$.

The torque controller enables optimal energy conversion [22] in Region 2 (below rated wind speed) using a look-up table, and, in general, holds the torque constant at the rated value $\tau_{\text{gen}} = 2.8$ Nm in Region 3 (above rated wind speed, where pitch control is active). A proportional-integral (PI) controller with saturation limits is used to handle the transition between partial and full load operation near the rated wind speed (Figure 6).

The PI controller aims to keep the rotor speed at 10 rpm below the rated value by setting the generator torque. When the rotor speed is much lower than the rated speed, the PI controller is saturated at the optimal torque $\tau^{\star}_{\text{gen}}(\Omega)$ (which varies with rotor speed), ensuring the optimal energy conversion. As the wind speed increases and the rotor speed approaches its rated value, the PI controller increases the torque and tries to keep the rotor speed at 10 rpm below the rated value. Finally, when the wind speed increases sufficiently, the PI controller saturates at the rated torque $\tau_{\text{gen,rated}} = 2.8$ Nm and the blade pitch controller takes over. This leads to a narrow region that is suboptimal in terms of energy conversion, but ensures a smooth transition between partial and full load conditions. An anti-windup scheme is used to prevent excessive accumulation of the integral term during saturation (not shown in Figure 6 for simplicity).

Since the transition between the full and partial load conditions is done based only on the rotor speed measurement, it is possible that the torque controller reacts to sudden drops in wind speed even when the pitch controller is still active. In other words, the torque controller will assist the pitch controller in keeping the rotor speed at the rated value during strong dips in wind speed, at the expense of an additional loss of power.

Note that the torque controller implemented is not designed specifically to work with the model predictive blade pitch controller and could be used with other pitch controllers.

D. Integrator Anti-windup for the Model Predictive Controller

Finally, to aid with transitions between the model predictive blade pitch controller (Section III-B) and the torque controller (Section III-C), we apply an anti-windup technique to prevent the integral term x_2 from growing unbounded when the blade pitch is saturated at the lower limit (during Region 2 operation). Instead of passing the rotor speed error to the integrator, zero is passed when the 'Region 2 operation' condition is met, as follows:

$$e(k) = \begin{cases} 0, & \beta(k-1) = \beta_{\text{min}} \text{ and } \Omega(k) < \Omega_{\text{nom}} \\ \delta\Omega(k), & \text{otherwise ,} \end{cases} \tag{11}$$

where $e(k)$ is the input to the integrator. We thus replace $\hat{x}_2 = \int \delta\Omega(k)$ with $\hat{x}_2 = \int e(k)$ when constructing the current state $\hat{x}(k)$.

Making this change means that the linear model (1) is no longer accurate for predictions (since it does not account for the anti-windup behavior). However, we consider this discrepancy simply as a part of the nonlinear behavior of the true system that our linearized model does not account for.

IV. Test Setup

The controller was tested on MoWiTO (see Section II) running in a wind tunnel at ForWind – Centre for Wind Energy Research at the University of Oldenburg. In all test cases, we compare the FB only configuration ($d_k \equiv 0 \, \forall \, k$) to the FB/FF case, where disturbance preview is enabled.

A. Controller Implementation

The controller described in Section III is implemented on a National Instruments CompactRIO-9066 controller (cRIO) that has a 667 MHz dual-core CPU. It is implemented as a LabVIEW virtual instrument (VI), which calls a compiled C++ shared library to run the MPC code. The C++ code used is essentially a problem-specific wrapper function that calls qpOASES [45], an open-source quadratic problem solver tailored towards MPC (see Section VI for details on solving the MPC problem). The compiled library is called from LabVIEW using a Call Library Function Node set to the 'Run in any thread' configuration.

B. Hardware-in-the-loop Testing

Following MATLAB/Simulink simulation testing of the control law (implemented using qpOASES's MATLAB interface [45]) and prior to full testing on MoWiTO, we carried out so-called 'hardware-in-the-loop' testing. The controller was implemented on the cRIO in its final form, but the physical plant is replaced by the FAST simulation model controlled by a tailor-made dynamic link library that provides communication between the cRIO and FAST running on a PC, developed by Syed Muzaher Hussain Shah at the University of Oldenburg. We use this as a valuable method for checking computational performance and stability prior to full testing on expensive equipment. In particular, we at this point determined that we needed to reduce our prediction horizon length from $N = 20$, which was used in our previous study [34], to $N = 10$ to achieve a solve time that we considered to be fast enough for our application (see Section VI). In previous work, we saw little difference in performance using such a reduction [42, Figure 5].

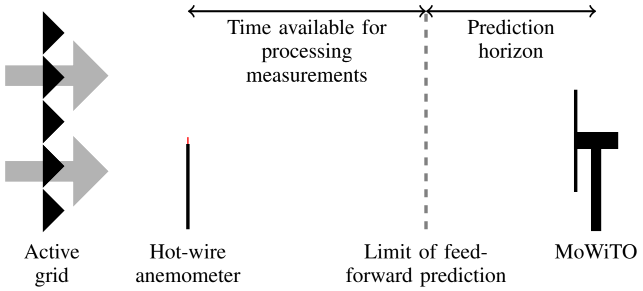

Time available for processing measurements
Prediction horizon
Active grid
Hot-wire anemometer
Limit of feed-forward prediction
MoWiTO

Fig. 7. Diagram of wind tunnel test setup. The active grid controls airflow into the tunnel test section; the hot-wire anemometer measures the wind speed; and MoWiTO responds to disturbances.

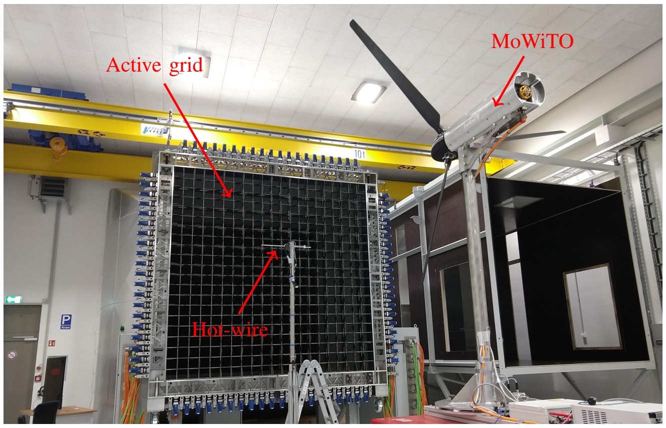


Fig. 8. Photograph of wind tunnel test setup with various components labelled (cf. Figure 7)

C. Test Layout

The physical layout of the experiment is shown in Figures 7 & 8. The experiment was carried out inside a closed-circuit wind tunnel with a 3 × 3 m^2 cross-sectional test section at ForWind – Center for Wind Energy Research at the University of Oldenburg. The 'active grid', described by Kröger et al. [37] and designed after Makita [46], [47], is capable of creating reproducible inflow conditions [48]. Different flows, from single velocity steps and gusts to atmospheric-like turbulence driven by stochastic processes, can be tailored [49] for turbine testing. As described in Section IV-E, gusts and turbulent inflow are used to test controller performance in this study.

D. Preview Measurements

In most studies investigating feedforward control of full-scale wind turbines, preview measurements are assumed to come from a lidar scanner [9], [26]. Verwaal et al. use a pitot tube measurement for their wind tunnel campaign [19]. In the present work, we generate preview wind speed measurements using a hot-wire anemometer situated 2.7 m upstream of the turbine (Figure 7) [34].

Preview wind measurements, whether generated by lidar or hot-wire, need to be filtered to remove both sensor noise and certain aspects of the wind disturbance signal before being used for feedforward control (Figure 5). In particular, there should be sufficient filtering to remove high-frequency turbulent eddies that evolve as they move downstream and have therefore changed by the time they reach the turbine rotor [50]. In addition, the filter helps to remove the small transversal components of velocity present in the hot-wire anemometer wind speed measurement. Finally, the preview measurement should be delayed to account for the time it takes for the measured wind field to move downstream to the turbine location [51]. The disturbance is assumed to advect downstream from the hot-wire anemometer to the turbine rotor at the mean wind speed with negligible distortion of the low-frequency components under Taylor's frozen turbulence hypothesis [52], which takes 38 time steps at 100 Hz based on the test setup and mean wind speed of 7.2 m/s.

Part of this required delay is acheived by filtering, which imparts a group delay, or lag, on the signal output. We use a moving average filter with $n_{\text{filt}} = 35$ samples, consistent with our previous work [34], since it imparts a frequency-independent delay of $(n_{\text{filt}}-1)/2 = 17$ time steps. As a consequence of the MPC prediction horizon, the filtered disturbance must be available for use $N = 10$ steps ahead of the turbine (and thus, as discussed in Sinner & Pao [42], the filter order $n_{\text{filt}} - 1$ and preview horizon length must be traded off so as not to exceed the total number of time steps available between the hot-wire location and the turbine). The remaining 11 time steps are accounted for by storing the filtered disturbance in a buffer before it enters the prediction horizon. The filter and buffer are implemented in LabVIEW. Note that the reduction in prediction horizon length (from 20 to 10) would have allowed for a higher filter order, but we decided to keep $n_{\text{filt}} = 35$ to be consistent (where possible) with our previous work [34].

A mathematically equivalent approach is to use a non-causal moving average filter with time-domain equation

$$d(k) = \frac{1}{n_{\text{filt}}} \sum_{j=1}^{n_{\text{filt}}} d_{\text{raw}} \left(k + \frac{n_{\text{filt}} + 1}{2} - j \right) \tag{12}$$

and transfer function

$$H(z) = \frac{D(z)}{D_{\text{raw}}(z)} = \frac{1}{n_{\text{filt}}} \frac{z^{(n_{\text{filt}}-1)} + z^{(n_{\text{filt}}-2)} + \cdots + z + 1}{z^{(n_{\text{filt}}-1)/2}}, \tag{13}$$

where d_{raw} is the raw hot-wire measurement shifted downstream to the turbine location, i.e. $d_{\text{raw}}(k) = v_{\text{meas}}(k - 38)$. Again, n_{filt} (which should be a positive, odd integer) is the number of samples in the moving average. This is the way we presented the filter in previous work [34], [42]. Examples comparing d to d_{raw} can be seen in Figure 9.

E. Test Conditions

We tested the controller in various wind conditions as listed in Table III, focusing on Region 3 (above-rated) and Region 2.5 (transition) winds, where the MPC produces interesting results. Tests T1–T3 use 'Mexican hat' gust profiles based on extreme operating gusts [53, pp. 33–34]. Tests T4 and T5 use (repeatable) turbulent sequences that show the behavior of atmospheric wind similar to, although not designed to precisely match, the normal turbulence model [53, pp. 31–32], which is considered representative of realistic turbulence. Each wind profile is generated by the active grid and has comparatively small fluctuations in the transversal inflow components. No shear is added to the inflow.

T1 (Figures 9a & 10a) is very similar to the tests used in Sinner et al. [34], and maintains above-rated operation throughout (as the blade pitch never reaches the minimum value $\beta_{\text{min}} = 0°$). This is used primarily to check that the controller is behaving as expected and producing similar results to the unconstrained case [34].

T2 (Figures 9b & 10b) is similar to T1, but uses a base wind speed that results in below-rated operation. As the gust impacts the turbine, the optimal blade pitch angle (as determined by the MPC) increases above zero and then reduces back to zero as the gust passes. The MPC is active for the duration of the test; however, when the wind speeds are low enough that the rotor is not reaching its rated speed, the optimal solution of the MPC is to hold the blades at their lower limit (10) to produce maximum aerodynamic torque (see Figure 4). In fact, the base wind speed of 7.2 m/s used in this test was expected to be an above-rated condition (see Section II-A), with expected drops into below-rated operation during the gust, but physical discrepancies resulted in base operation marginally below the rated rotor speed of 480 rpm.

The gust used for T3 (Figures 9c & 10c) is shorter in duration than that of T2. Here, we aim to activate the blade pitch rate constraint (9) by producing a rapid change in the rotor speed. Note that the predicted disturbance leads the true disturbance in this test. This is due to the discrepancy between the nominal wind speed assumed for preview advancing (7.2 m/s) and the actual (lower) mean wind speed for the test. This issue would need to be addressed for a full implementation of feedforward control over a wide range of wind speeds (see Section VII).

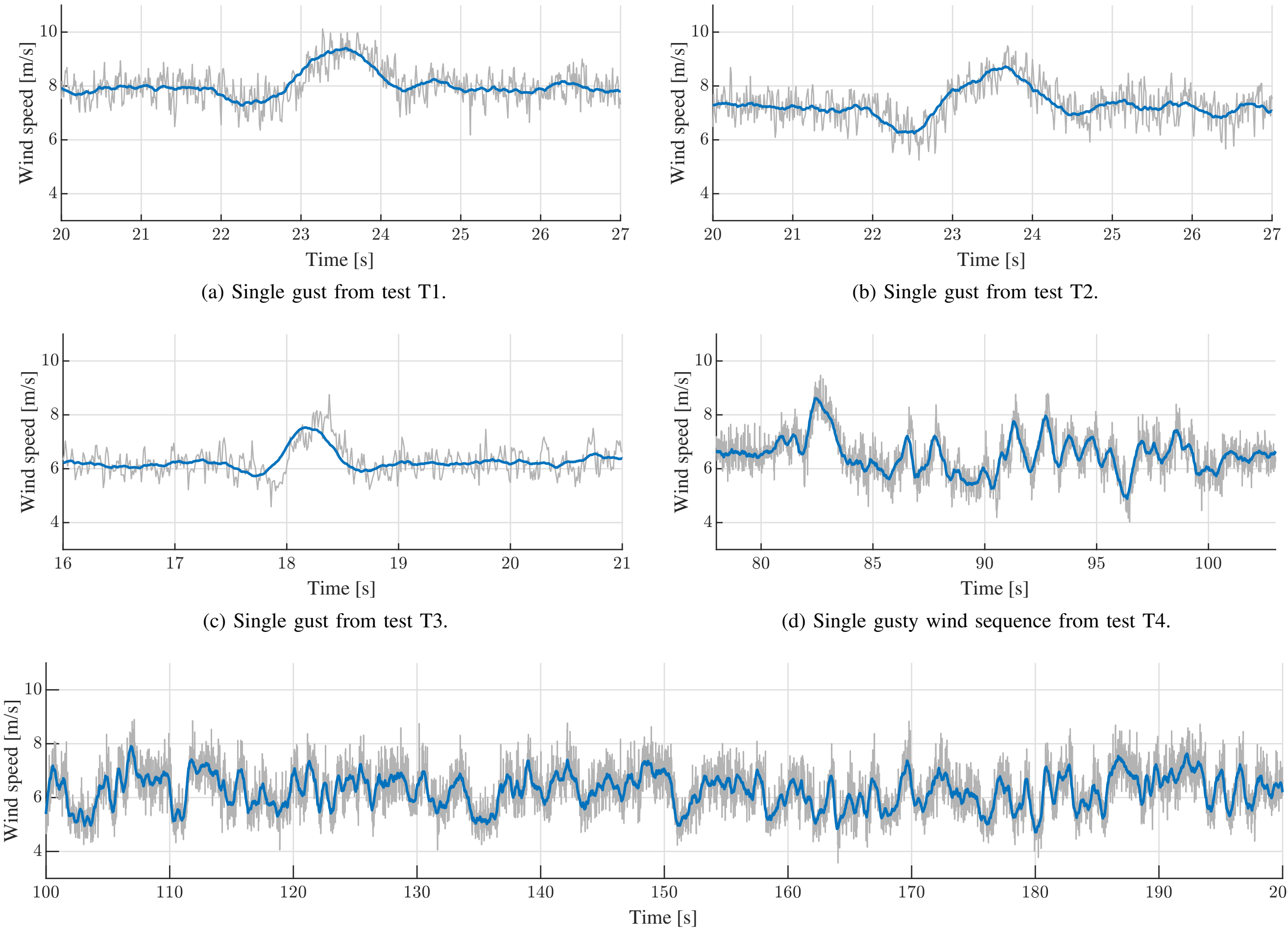
Although single deterministic gusts are useful for verifying controller performance, they are not particularly representative of real wind fields. T4 and T5 (Figures 9d & 10d and 9e & 10e, respectively) present more realistic scenarios. T4 uses a series of turbulent gusts of varying magnitudes and durations, whereas T5 uses a persistent turbulent wind profile, with turbulence intensities of approximately 18% and 15%, respectively, at the turbine location. Both require frequent transition between above- and below-rated operation, testing the controllers' ability to solve the MPC problem in the presence of constraints that are frequently being activated and deactivated.

V. Results

We consider the main result of this paper to be the successful deployment of MPC for blade pitch control running at the controller sample rate of a modern utility-scale wind turbine

TABLE III
Description of Test Sequences

Test	Type	Repetitions	Base wind speed [m/s]	Duration [s]	Spacing [s]	Figure references
T1	'Mexican hat' gust	10	7.9	4	7	9a, 10a
T2	'Mexican hat' gust	10	7.2	4	7	9b, 10b
T3	'Mexican hat' gust	10	6.3	1	5	9c, 10c
T4	Gusty sequence	10	6.6	20	25	9d, 10d
T5	Persistent turbulence	1	6.1	540	NA	9e, 10e



(a) Single gust from test T1.

(b) Single gust from test T2.

(c) Single gust from test T3.

(d) Single gusty wind sequence from test T4.

(e) 100 s taken from the 9-minute record used for test T5.

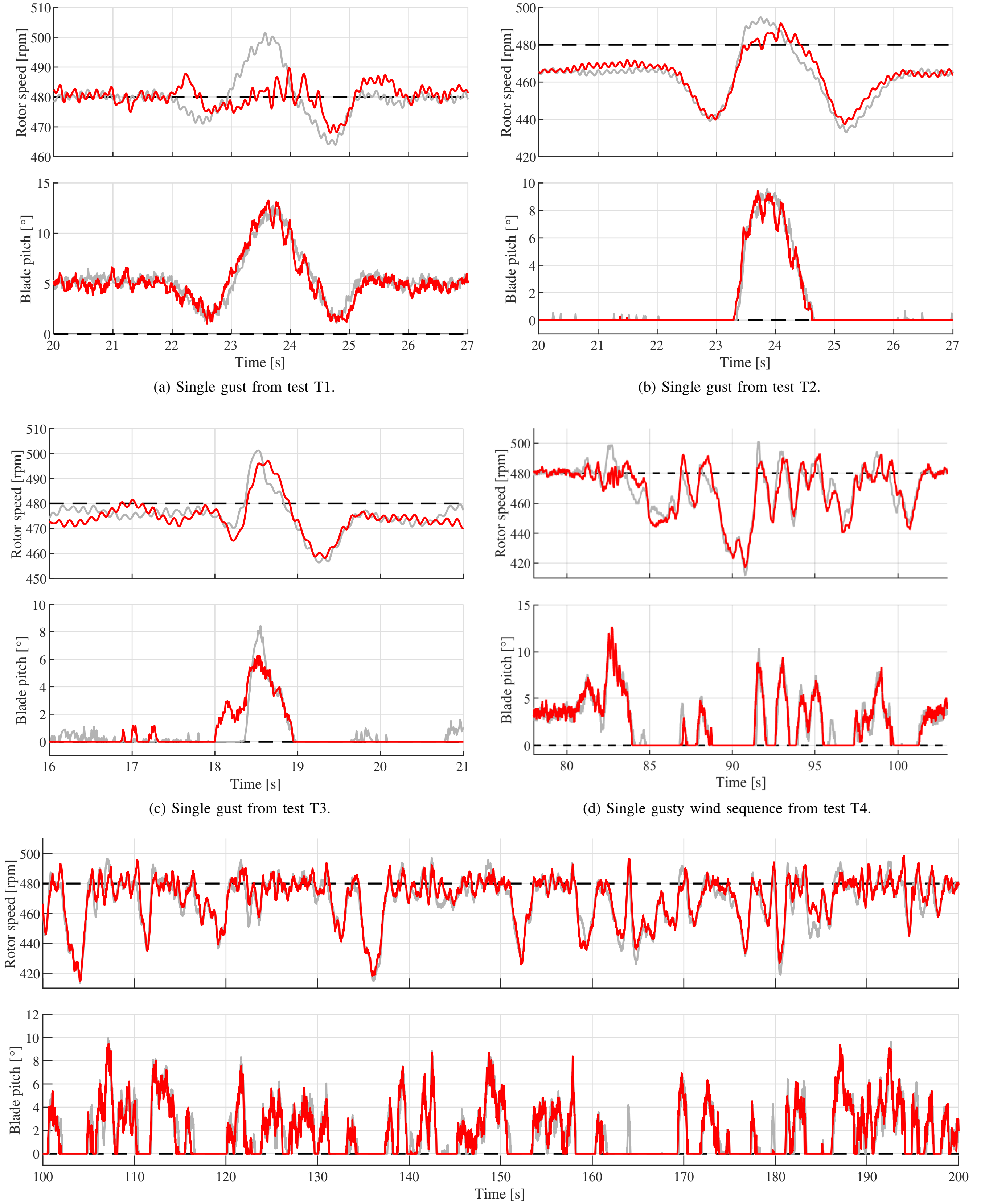
Fig. 9. Excerpts of wind speed as measured by the hot-wire anemometer for the various test cases described in Section IV-E. The gray signal is the 'raw' signal d_{raw}, which is the hot-wire measurement v_{meas} shifted downstream to the turbine location, while the blue signal is the filtered version d that is eventually used by the MPC in the FB/FF configuration (see Section IV-D for details).

(the NREL 5MW turbine, for example, uses a sample rate of 80 Hz [38]), in both feedback only and feedback/feedforward configurations. Figure 10 presents time-series results from tests T1–T5, and are consistent with the expected behavior of a wind turbine in transition and above-rated winds.

TABLE IV
Performance Improvements using Feedforward Control

Test	RMS $\left(\delta\Omega_{\geq 0}\right)$	RMS $\left(\dot{\beta}\right)$
T1	61.4%	-2.9%
T2	52.6%	4.6%
T3	5.6%	14.5%
T4	21.0%	2.8%
T5	5.0%	2.3%

Statistical results from all tests are presented in Table IV. Of particular interest in this work are rotor speed regulation in above-rated winds and the level of blade pitch actuation required. Because it is difficult to determine, in transition winds, whether the turbine is operating above- or below-rated, we consider the root-mean-squared (RMS) value of the rotor speed deviation taken over times when the deviation is positive (the rotor speed is above its rated value). That is, we remove



(a) Single gust from test T1.

(b) Single gust from test T2.

(c) Single gust from test T3.

(d) Single gusty wind sequence from test T4.

(e) 100 s taken from the 9-minute record used for test T5.

Fig. 10. Excerpts of rotor speed (top plots) and blade pitch signals (lower plots) for the various test cases described in Section IV-E. The red signal is the FB/FF configuration, while the gray signal is the FB only configuration.

all data from times when the rotor speed is below rated speed before finding the RMS value of the remaining data, denoted $\delta\Omega_{\geq 0}$. We admit that this does not give the complete picture, since times when the wind is high enough but the rotor speed is below rated (for example, 24–25 s in Fig 10a, top) are removed from analysis, but we consider this a reasonable price to pay to ensure we are analyzing purely above-rated operation, where the explicit goal of the controller is to regulate the rotor speed to rated. When considering blade pitch actuation, on the other hand, we use the entire time series (including times when the rotor speed is significantly below rated) when calculating the RMS blade pitch rate $\dot{\beta}$. This is because times when one controller is pitching and the other is not (for example, 19–21 s in Fig 10c, bottom) are important to include, and pitching during below-rated operation (e.g. 20–23 s in Fig 10b, bottom) should be penalized.

To investigate the improvement (or degradation) in performance when utilizing feedforward measurements, the statistics we present in Table IV are in the form of a percentage reduction from the FB only to the FB/FF case. Overall, the availability of feedforward measurements appears to be having the desired affect: we see moderate to significant reductions in rotor speed deviation above the rated speed, while, in most cases, reducing blade pitching.

An interesting aspect of these findings is that, considering T2–T5 Figures 10b–10e (bottom), most of the reduction in blade pitching appears to result from the FB/FF controller maintaining zero pitch at times when the FB only controller makes small deviations into above-rated operation. This could be explained by the knowledge of the future disturbance trajectory for the FB/FF case, which can indicate whether an increase in wind speed is only temporary (e.g. 19–21 s in Fig 10c) or sustained. In the latter case, earlier (and less aggressive) actuation in anticipation of significant future events (e.g. 18–19 s in Fig 10c) may be achieved. On the other hand, in T1 (Figure 10a) where the entire test takes place in above-rated operation, we see a slight increase in blade pitching from the FB only to the FB/FF case. This can likely be explained by noise injected into the control signal u by the disturbance measurement d, and could be less of an issue for feedforward measurements provided by lidar for a utility-scale wind turbine [34].

Finally, we point out that the rotor speed and blade pitch angle for T1 (Figure 10a top and bottom, respectively) follow very similar trajectories to our previous tests of unconstrained optimal control [34, Figure 3]. In particular, the FB/FF MPC achieves a reduction in RMS rotor speed error of 61.4% compared to the FB only MPC across the entire sequence, similar to the 54.0% reduction in standard deviation seen previously when comparing unconstrained FB/FF to unconstrained FB only during the gust event[1] [34, Table 2].

VI. Analysis of Computational Performance

Having demonstrated the viability of MPC for blade pitch control, we now turn our attention to the computational performance of the implementation, given that MPC is a relatively computationally demanding control approach. While the analysis provided here is by no means exhaustive, we hope that it will be of practical benefit to others interested in physical implementations of MPC for wind turbines and other systems. As mentioned in Section IV-A, the MPC is implemented as a compiled C++ library called from a LabVIEW VI. The critical piece of the compiled library is qpOASES, an open-source quadratic program solver developed by Ferreau et al. [45].

From a host of possible algorithms and software packages for solving quadratic programs (again noting that the linear MPC problems (5), (6), and (7) are quadratic programs), we chose qpOASES for several reasons. qpOASES is tailored for MPC applications in that it uses the efficient active set algorithm, which can be warm started, and has a method for terminating the algorithm early and providing an intermediary solution in case the time available for computation expires. Further, qpOASES is readily implementable: while the primary distribution is for C++, there is also a MATLAB interface [45], which we found convenient for proof-of-concept simulation testing. Finally (likely as a result of the above), qpOASES has been used for MPC in the wind turbine literature where real-time behavior was desired [35], and in particular was used by Verwaal et al. [19].

Roughly speaking, the active set method employed by qpOASES uses the solution from the previous time step, and ignores all inactive constraints (that is, rows of the constraint (7b)) when finding the optimal solution. As the problem parameters $(\hat{x}(k), \mathbf{d}(k))$ change, constraints may become active (are added to the active set) or inactive (are removed from the active set). When this occurs, the algorithm adds (removes) the constraint from the 'working set' before completing another iteration. It is possible for several iterations—or 'working set recalculations'—to occur before the optimal solution, and corresponding active set, is found. For details on the active set method, see Ferreau et al. [45], [54].

As a consequence of requiring varying numbers of working set recalculations per controller time step, the time to compute the optimal solution can vary significantly. In our implementation, we required a controller sampling frequency of 100 Hz—however, we reduced our allowable computation time to $T_{\text{solve}} = 3$ ms to allow time for communication and other protocols within the 10 ms sampling time. qpOASES was designed to prevent a further working set recalculation, and provide a suboptimal solution, in the case that the next recalculation would violate the allowable computation time. While sacrificing optimality in the case of significant changes in the active set, this ensures real-time capabilities. The solution returned in this case is the solution of a quadratic program with a parameter $(\hat{x}(k), \mathbf{d}(k))'$ between the current parameter $(\hat{x}(k), \mathbf{d}(k))$ and the previous one $(\hat{x}(k-1), \mathbf{d}(k-1))$. See Ferreau et al. [54, Section 3.5] for details.

We demonstrate graphically the computational performance of the MPC in Figure 11. For the bottom plots, we use 'C' to denote that the qpOASES 'completed' (successfully found the optimal solution), 'T' to denote that qpOASES timed out and returned a suboptimal value (as described above), and 'F' to denote that qpOASES failed (for example, could not find

[1] The only mathematical difference between the RMS error and the standard deviation being that the RMS error takes into account bias in the signal, while the standard deviation does not.

a feasible solution). Note that, by our construction, the MPC problem (7), with the constraints described in Section III-B2 and parameters in Table II, is always feasible as long as $\beta(k-1) \not< \beta_{\text{min}}$, since $\mathbf{u} = \mathbf{0}$ is always a feasible point. For completeness, we included logic to set the blade pitch angle command to the solution of the corresponding unconstrained problem [34] in the case of a failed optimization. However, we did not encounter such a failure at any time during our testing.

Figure 11a shows the time required to return the control value $u(0)$, the number of working set recalculations required, and the solve status for a single gust in test T3 (cf. Figure 10c) for visualization purposes. Qualitatively, we see that there is an approximate minimum computation time associated with requiring zero work set recalculations, and that the number of recalculations appears to be lower in the region from 18–19 s, where the general affine constraint (10) is inactive but the simpler box constraint (9) may be (qpOASES has special handling for simple box constraints).

Figure 11b presents analysis of the computational performance metrics during turbulent inflow in test T5, based on the frequency of occurrence for each metric. The top plot (computation time, which is a continuous) has been binned. The 'floor' computation time appears to be between 0.5–0.7 ms, with the FB only case generally requiring less computation time when no working set recalculations are made. Interestingly, the FB/FF configuration completes with zero working set recalculations more often than the FB only configuration, but also may more often need a higher number of recalculations—and the completion rate (C) of the FB only controller is higher, at 98.0%, than that of the FB/FF controller, which timed out (T) 8.2% of the time for T5.

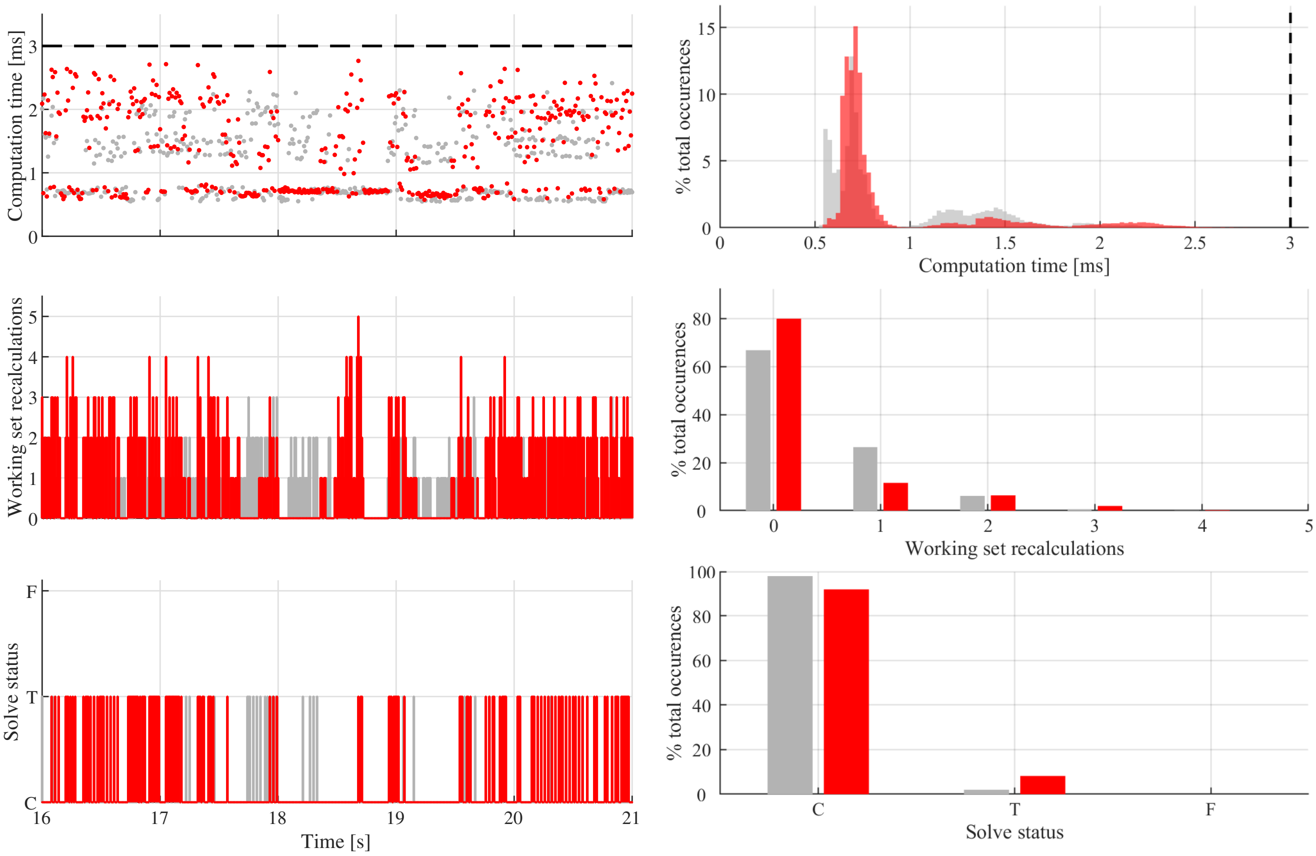
VII. Conclusions and Future Work

We consider the demonstration of MPC for blade pitch control on an experimental scaled wind turbine model to be the main contribution of this work. We now have further evidence that disturbance measurements can be used to improve turbine performance while maintaining similar levels of blade pitch activity. Similar to our previous study, the objective function (7a) for the optimal control sequence is unchanged between the feedback only and the feedback/feedforward case [34], such that the inclusion of feedforward action can be examined while holding other aspects of the controller constant. However, the presence of constraints (7b) in the condensed MPC problem (7) means that the optimal control $u(k)$ is no longer the sum of a linear feedback term (a gain on $\hat{x}(k)$) and a linear feedforward term (a gain on $\mathbf{d}(k)$) [34], but instead a nonlinear function of the input parameter $(\hat{x}(k), \mathbf{d}(k))$ whose value depends on solving a quadratic program online.

A major barrier to deployment of MPC in practice is the computational burden posed. In this study, we were able to run MPC at a rate of 100 Hz using a 3 ms limit on the time available to solve the MPC problem. The strict upper limit on solve time does result in the optimization algorithm terminating prematurely occasionally, but we did not see any evidence of detrimental performance as a result. The achievable solve time for other implementations will depend strongly on the number of constraints r at each time step, the prediction horizon length N, and the available hardware. However, we did see anecdotal evidence that the inclusion of disturbance preview information, while not changing the problem size (see Section III-B3), could result in a higher occurrence rate of longer solve times. The reason for this is unclear, but one possibility is that the inclusion of (noisy) preview measurements causes the solution trajectories to move around more than using only the (relatively clean) state as an input. If this is the case, then the solution to an approximate quadratic program may be no worse than the solution to the true one, in the sense that the difference between the two could be a manifestation of noise [54]. The work presented here has the potential to inform future implementations of MPC and other feedforward control techniques not only for wind turbine applications, but also in other fields where practical considerations (model accuracy, measurement noise, computation time, etc.) could pose barriers. In this work we have not aimed to study closely the sensitivity of the solve time to various aspects of the MPC problem, nor spent effort looking for the fastest possible solver. Rather, we have demonstrated that a readily available solver for MPC problems, qpOASES [45], provides performance that is satisfactory for our needs.

A few important issues have not been addressed in this work. In particular, we have used a linear time-invariant, single degree-of-freedom model (4) to represent all of above-rated wind turbine operation, which is a considerable simplification. The state dimension can be increased to include dynamics such as tower and blade bending without increasing the size of the condensed quadratic program (7); however, a more complex choice of state will likely require an observer to produce a state estimate $\hat{x}(k)$. This would allow turbine structural loads to be considered in the MPC cost function. Similarly, the extension from a linear time-invariant model to a linear parameter-varying one, used in much of the MPC literature for wind turbines [42], does not change the size of the problem (7) but would make the matrices $\mathbf{H}$ and $\mathbf{G}$ time-varying, which in turn means that some of the performance-enhancing aspects of qpOASES cannot be used. In this work, $\mathbf{h}$ and $\mathbf{g}$ are time-varying, but qpOASES is designed to handle this situation [45]. Further, model uncertainty was not considered, and a physical demonstration of a robust MPC approach for wind turbines has yet to appear in the literature.

Finally, we assume that disturbances move downstream at a fixed wind speed of $v_{\text{nom}} = 7.2$ m/s for all tests. To make the best use of feedforward information, more attention should be given to the precise delay, which could depend not only on the mean wind speed but also the induction zone in front of the turbine rotor [51] and tower bending. Similarly, varying the amount of filtering applied to the hot-wire measurment and in turn, the upstream distance of the hot-wire anemometer (while bearing in mind pitch actuation limitations), has not been addressed here. We had hoped to investigate some of these effects during this study, but were unable to complete testing due to time constraints in the duration of our testing window in the wind tunnel facility.



(a) Time series of computational performance metrics for T3.

(b) Relative frequency of computational performance metrics over T5.

Fig. 11. Computational performance metrics of the MPC for tests T3 (in time domain) and T5 (relative frequency of occurrences). The red signal is the FB/FF configuration and the gray signal is the FB only case. The upper limit on allowable computation time is represented by the black dashed line in the top plots.

Appendix
Matrix Definitions

The vectors and matrices defined for the matrix-vector form of the MPC problem (6) follow.

A. Spatial Vectors

$$\mathbf{x} \overset{\text{def}}{=} \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N) \end{bmatrix}, \mathbf{u} \overset{\text{def}}{=} \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-1) \end{bmatrix}, \mathbf{d}(k) \overset{\text{def}}{=} \begin{bmatrix} d(k) \\ d(k+1) \\ \vdots \\ d(k+N-1) \end{bmatrix}.$$

B. Model

$$\mathbf{A} \overset{\text{def}}{=} \begin{bmatrix} I \\ A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}, \mathbf{B} \overset{\text{def}}{=} \begin{bmatrix} 0 & \cdots & & 0 \\ B & 0 & & \vdots \\ AB & B & \ddots & \\ \vdots & & \ddots & 0 \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix},$$

$$\mathbf{B}_d \overset{\text{def}}{=} \begin{bmatrix} 0 & \cdots & & 0 \\ B_d & 0 & & \vdots \\ AB_d & B_d & \ddots & \\ \vdots & & \ddots & 0 \\ A^{N-1}B_d & A^{N-2}B_d & \cdots & B_d \end{bmatrix}.$$

C. Constraints

$$\mathbf{J} \overset{\text{def}}{=} \begin{bmatrix} J & 0 & \cdots & & 0 \\ 0 & J & \ddots & & \vdots \\ \vdots & & \ddots & & 0 \\ 0 & \cdots & 0 & J & 0 \end{bmatrix}, \mathbf{E} \overset{\text{def}}{=} \begin{bmatrix} E & 0 & \cdots & 0 \\ 0 & E & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & E \end{bmatrix},$$

$$\boldsymbol{\ell} \overset{\text{def}}{=} \begin{bmatrix} \ell \\ \ell \\ \vdots \\ \ell \end{bmatrix}.$$

D. Cost Function

$$\mathbf{Q} \overset{\text{def}}{=} \begin{bmatrix} Q & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & Q & 0 \\ 0 & \cdots & 0 & P \end{bmatrix}, \mathbf{R} \overset{\text{def}}{=} \begin{bmatrix} R & 0 & \cdots & 0 \\ 0 & R & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & R \end{bmatrix}.$$

Acknowledgment

We thank the German Academic Exchange Service (DAAD), who provided travel funding for this work; the Hanse-Wissenschaftskolleg in Delmenhorst, Germany; a Palmer Endowed Chair at the University of Colorado Boulder; and the National Renewable Energy Laboratory for ongoing support. We also acknowledge the state of Lower Saxony, which supported the wind tunnel campaign, active grid, and MoWiTO under the project "ventus efficiens".

References

[1] L. Henriksen, "Model predictive control of a wind turbine," Master's thesis, Technical University of Denmark, Kongens Lyngby, Denmark, 2007.

[2] D. Q. Dang, Y. Wang, and W. Cai, "Nonlinear model predictive control (NMPC) of fixed pitch variable speed wind turbine," in *Proceedings of the IEEE International Conference on Sustainable Energy Technologies*, 2008, pp. 29–33.

[3] A. Kusiak, Z. Song, and H. Zheng, "Anticipatory control of wind turbines with data-driven predictive models," *IEEE Transactions on Energy Conversion*, vol. 24, no. 3, pp. 766–774, 2009.

[4] L. Henriksen, N. Poulsen, and M. Hansen, "Nonlinear model predictive control of a simplified wind turbine," *Proceedings of the IFAC World Congress*, vol. 44, no. 1, pp. 551–556, 2011.

[5] M. Soltani, R. Wisniewski, P. Brath, and S. Boyd, "Load reduction of wind turbines using receding horizon control," in *Proceedings of the IEEE International Conference on Control Applications*, Denver, CO, Sep. 2011, pp. 852–857.

[6] J. H. Laks, "Preview scheduled model predictive control for horizontal axis wind turbines," Ph.D. dissertation, University of Colorado, Boulder, CO, 2012.

[7] M. Spencer, K. Stol, C. Unsworth, J. Cater, and S. Norris, "Model predictive control of a wind turbine using short-term wind field predictions," *Wind Energy*, vol. 16, no. 3, pp. 417–434, 2012.

[8] M. Mirzaei, L. Henriksen, N. Poulsen, H. Niemann, and M. Hansen, "Individual pitch control using LIDAR measurements," in *Proceedings of the IEEE International Conference on Control Applications*, Dubrovnik, Croatia, Oct. 2012, pp. 1646–1651.

[9] M. Harris, M. Hand, and A. Wright, "Lidar for turbine control," NREL, Golden, CO, Tech. Rep. NREL/TP-500-39154, Jan. 2005.

[10] A. Koerber and R. King, "Combined feedback-feedforward control of wind turbines using state-constrained model predictive control," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 4, pp. 1117–1128, Jul. 2013.

[11] D. Schlipf, P. Grau, S. Raach, R. Duraiski, J. Trierweiler, and P. W. Cheng, "Comparison of linear and nonlinear model predictive control of wind turbines using LIDAR," in *Proceedings of the American Control Conference*, Portland, OR, Jun. 2014, pp. 3742–3747.

[12] D. Schlipf, D. Schlipf, and M. Kühn, "Nonlinear model predictive control of wind turbines using LIDAR," *Wind Energy*, vol. 16, no. 7, pp. 1107–1129, Oct. 2013.

[13] E. Tofighi, T. Faulwasser, and C. M. Kellet, "Nonlinear model predictive control approach for structural load mitigation of wind turbines in presence of wind measurement uncertainties," in *Proceedings of the Australian Control Conference*, Gold Coast, Australia, Nov. 2015, pp. 210–214.

[14] S. Gros, "An economic NMPC formulation for wind turbine control," in *Proceedings of the Conference on Decision and Control*, Florence, Italy, 2013, pp. 1001–1006.

[15] S. Gros and A. Schild, "Real-time economic nonlinear model predictive control for wind turbine control," *International Journal of Control*, vol. 90, no. 12, pp. 2799–2812, 2017.

[16] M. Mirzaei, N. K. Poulsen, and H. H. Niemann, "Robust model predictive control of a wind turbine," in *Proceedings of the American Control Conference*, Montrseal, Canada, Jun. 2012, pp. 4393–4398.

[17] W. Lio, B. Jones, and J. Rossiter, "Preview predictive control layer design based upon known wind turbine blade-pitch controllers," *Wind Energy*, vol. 20, no. 7, pp. 1207–1226, Feb. 2017.

[18] M. Mirzaei, M. Soltani, N. K. Poulsen, and H. H. Niemann, "Model predictive control of wind turbines using uncertain LIDAR measurements," in *Proceedings of the American Control Conference*, Washington, D.C., Jun. 2013, pp. 2235–2240.

[19] N. Verwaal, G. van der Veen, and J.-W. van Wingerden, "Predictive control of an experimental wind turbine using preview wind speed measurements," *Wind Energy*, vol. 18, no. 3, pp. 385–398, Mar. 2015.

[20] D. Castaignet, I. Couchman, N. K. Poulsen, T. Buhl, and J. J. Wedel-Heinen, "Frequency-weighted model predictive control of trailing edge flaps on a wind turbine blade," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 4, pp. 1105–1116, 2013.

[21] M. Sinner and L. Y. Pao, "A comparison of individual and collective pitch model predictive controllers for wind turbines," in *Proceedings of the American Control Conference*, Milwaukee, WI, Jun. 2018, pp. 1509–1514.

[22] L. Pao and K. Johnson, "Control of wind turbines," *IEEE Control Systems Magazine*, pp. 44–62, Apr. 2011.

[23] L. C. Henriksen, M. H. Hansen, and N. K. Poulsen, "Wind turbine control with constraint handling: a model predictive control approach," *IET Control Theory Applications*, vol. 6, no. 11, pp. 1722–1734, 2012.

[24] C. Bottasso, P. Pizzinelli, C. Riboldi, and L. Tasca, "LiDAR-enabled model predictive control of wind turbines with real-time capabilities," *Renewable Energy*, vol. 71, pp. 442–452, Nov. 2014.

[25] A. Jain, G. Schildbach, L. Fagiano, and M. Morari, "On the design and tuning of linear model predictive control for wind turbines," *Renewable Energy*, vol. 80, pp. 664–673, Aug. 2015.

[26] A. Scholbrock, P. Fleming, D. Schlipf, A. Wright, K. Johnson, and N. Wang, "Lidar-enhanced wind turbine control: Past, present, and future," in *Proceedings of the American Control Conference*, Boston, MA, Jul. 2016, pp. 1399–1406.

[27] S. P. Mulders, T. G. Hovgaard, J. D. Grunnet, and J.-W. van Wingerden, "Preventing wind turbine tower natural frequency excitation with a quasi-LPV model predictive control scheme," *Wind Energy*, vol. 23, no. 3, pp. 627–644, 2020.

[28] W. H. Lio, J. A. Rossiter, and B. L. Jones, "A review on applications of model predictive control to wind turbines," in *UKACC International Conference on Control*, Loughborough, UK, 2014, pp. 673–678.

[29] A. Scholbrock, P. Fleming, L. Fingersh, A. Wright, D. Schlipf, F. Haizmann, and F. Belen, "Field testing LIDAR based feed-forward controls on the NREL controls advanced research turbine," in *AIAA Aerospace Sciences Meeting*, Jan. 2013.

[30] D. Schlipf, P. Fleming, F. Haizmann, A. Scholbrock, M. Hofsäß, A. Wright, and P. Cheng, "Field testing of feedforward collective pitch control on the CART2 using a nacelle-based lidar scanner," in *Journal of Physics: Conference Series*, vol. 555 (012090), no. 1, 2014.

[31] F. Haizmann, D. Schlipf, S. Raach, A. Scholbrock, A. Wright, C. Slinger, J. Medley, M. Harris, E. Bossanyi, and P. Cheng, "Optimization of a feed-forward controller using a CW-lidar system on the CART3,"

in *Proceedings of the American Control Conference*, Chicago, IL, Jul. 2015, pp. 3715–3720.

[32] D. Schlipf, P. Fleming, S. Raach, A. Scholbrock, F. Haizmann, R. Krishnamurthy, M. Boquet, and P. Cheng, “An adaptive data processing technique for lidar-assisted control to bridge the gap between lidar systems for wind turbines,” in *Proceedings of the EWEA Annual Event*, Paris, France, Nov. 2015.

[33] A. A. Kumar, E. A. Bossanyi, A. K. Scholbrock, P. Fleming, M. Boquet, and R. Krishnamurthy, “Field testing of LIDAR assisted feedforward control algorithms for improved speed control and fatigue load reduction on a 600 kW wind turbine,” in *Proceedings of the EWEA Annual Event*, Paris, France, Nov. 2015.

[34] M. Sinner, V. Petrović, F. Berger, L. Neuhaus, M. Kühn, and L. Y. Pao, “Wind tunnel testing of an optimal feedback/feedforward control law for wind turbines,” in *Proceedings of the IFAC World Congress*, Berlin, Germany, Jul. 2020.

[35] S. Gros, M. Vukov, and M. Diehl, “A real-time MHE and NMPC scheme for wind turbine control,” in *Proceedings of the IEEE Conference on Decision and Control*, Florence, Italy, Dec. 2013, pp. 1007–1012.

[36] F. Berger, L. Kröger, D. Onnen, V. Petrović, and M. Kühn, “Scaled wind turbine setup in a turbulent wind tunnel,” in *Journal of Physics: Conference Series*, vol. 1104 (012026), no. 1, Jan. 2018.

[37] L. Kröger, J. Frederik, J.-W. van Wingerden, J. Peinke, and M. Hölling, “Generation of user defined turbulent inflow conditions by an active grid for validation experiments,” in *Journal of Physics: Conference Series*, vol. 1037 (052002), no. 5, Jan. 2018.

[38] J. Jonkman, S. Butterfield, W. Musial, and G. Scott, “Definition of a 5-MW reference wind turbine for offshore system development,” NREL, Golden, CO, Tech. Rep. NREL/TP-500-38060, Feb. 2009.

[39] J. Jonkman and M. Buhl Jr., “FAST user’s guide,” NREL, Golden, CO, Tech. Rep. NREL/EL-500-38230, Aug. 2005.

[40] B. Jonkman and J. Jonkman, “FAST v8.16.00a-bjj,” Jul. 2016, available: wind.nrel.gov/nwtc/docs/README_FAST8.pdf.

[41] B. Jonkman, “TurbSim user’s guide: Version 1.50,” NREL, Golden, CO, Tech. Rep., Sep. 2009.

[42] M. Sinner and L. Pao, “A study on horizon length for preview-enabled model predictive control of wind turbines,” in *Proceedings of the American Control Conference*, Philadelphia, PA, Jul. 2019, pp. 3488–3493.

[43] G. Franklin, D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 8th ed. Pearson, 2019.

[44] J. Rawlings, “Tutorial overview of model predictive control,” *IEEE Control Systems Magazine*, vol. 20, no. 3, pp. 38–52, 2000.

[45] H. Ferreau, C. Kirches, A. Potschka, H. Bock, and M. Diehl, “qpOASES: A parametric active-set algorithm for quadratic programming,” *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.

[46] H. Makita, “Realization of a large-scale turbulence field in a small wind tunnel,” *Fluid Dynamics Research*, vol. 8, no. 1, pp. 53–64, 1991.

[47] P. Knebel, A. Kittel, and J. Peinke, “Atmospheric wind field conditions generated by active grids,” *Experiments in Fluids*, vol. 51, pp. 471–481, 2011.

[48] L. Kröger, L. Neuhaus, J. Peinke, G. Gülker, and M. Hölling, “Turbulence generation by active grids,” in *Progress in Turbulence VIII*, R. Örlü, A. Talamelli, J. Peinke, and M. Oberlack, Eds. Cham: Springer International Publishing, 2019, pp. 191–196.

[49] L. Neuhaus, F. Berger, J. Peinke, and M. Hölling, “Exploring the capabilities of active grids,” *submitted to Experiments in Fluids*, (under review).

[50] E. Simley and L. Pao, “A longitudinal spatial coherence model for wind evolution based on large-eddy simulation,” in *Proceedings of the American Control Conference*, Chicago, IL, Jul. 2015, pp. 3708–3714.

[51] F. Dunne, L. Pao, D. Schlipf, and A. Scholbrock, “Importance of lidar measurement timing accuracy for wind turbine control,” in *Proceedings of the American Control Conference*, Portland, OR, Jun. 2014, pp. 3716–3721.

[52] D. Schlipf, D. Trabucchi, O. Bischoff, M. Hofsäß, J. Mann, T. Mikkelsen, A. Rettenmeier, J. J. Trujillo, and M. Kühn, “Testing of frozen turbulence hypothesis for wind turbine applications with a scanning lidar system,” in *Proceedings of the International Symposium for the Advancement of Boundary Layer Remote Sensing*, Paris, France, 2010.

[53] “Wind energy generation systems – Part 1: Design requirements,” International Electrotechnical Commission, Geneva, Switzerland, Standard IEC 61400-1, Edition 4.0, 2019.

[54] H. Ferreau, H. Bock, and M. Diehl, “An online active set strategy to overcome the limitations of explicit MPC,” *International Journal of Robust and Nonlinear Control*, vol. 18, no. 8, pp. 816–830, 2008.