Computationally efficient model predictive control of complex wind turbine models

Evans, Martin A.; Lio, Wai Hou

Published in:
Wind Energy

Link to article, DOI:
10.1002/we.2695

Publication date:
2022

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.
Computationally efficient model predictive control of complex wind turbine models

Martin A. Evans1 | Wai Hou Lio2

1DNV, Bristol, UK

2Department of Wind Energy, Technical University of Denmark (DTU), Roskilde, Denmark

Correspondence
Martin A. Evans, DNV, One Linear Park, Avon Street, Bristol BS2 0PS, UK.
Email: martin.evans@dnv.com

Funding information
European Union’s Horizon 2020, Grant/Award Number: 727680

Abstract
As wind turbines are designed with longer blades and towers, it becomes increasingly important to factor structural modes into the design of the controller. In classical turbine controllers, where pitch-speed, torque-speed, drivetrain and tower dampers are designed separately, it has for years been commonplace to base that design on a linearisation of the existing high-fidelity aeroelastic model. Furthermore, any measurement filters that are required at run-time are included in the control loop shaping process. In contrast, most previous work on model predictive control (MPC) for wind turbines uses simplified models and ignores the need or effect of measurement filters. In this work, we demonstrate a mostly automatic design process that takes a detailed linearised model from an aeroelastic simulation package and adds linear filters and feedback, to produce a model predictive controller with low run-time computational complexity. The tuning process is substantially simpler than classical control, making it an attractive tool in industrial applications.

KEYWORDS
exponential basis functions, linearisation, MPC, wind turbine control

1 | INTRODUCTION

Due to the increasing size and flexibility of modern wind turbine designs, more blade and tower vibrational modes lie within the controller bandwidth, meaning those modes must be included in the model on which the controller is designed. The industry standard process to design the controller is to linearise the full aeroelastic model of the turbine and to carefully perform loop shaping with PID and filters for each feedback loop separately. Input constraints are accommodated via anti-windup strategies, but no constraints on the system outputs are guaranteed.

Model predictive control (MPC), in contrast, is an optimisation subject to constraints on inputs and outputs. Like the linear-quadratic regulator (LQR), it has an intuitive tuning process where the trade-off between control activity and output deviations is defined as a quadratic cost function.

Minimisation of that quadratic cost subject to linear constraints is known as a quadratic programme (QP), which can be rapidly solved online on even a modest computer, provided the number of degrees of freedom is kept low. MPC makes predictions of the future states of the plant in terms of those degrees of freedom and the current estimated state. With a linear prediction model and linear constraints, the optimal control action to apply to the plant at each time step can be found by solving a single QP.

Rather than building the QP online, starting from the estimated state and multiplying by the dynamical system matrices for each prediction step in the horizon, the QP is formed offline with the estimated state as a variable. It is not an optimisation variable—it’s value is known at run-time and is applied to the QP as an equality constraint. This allows prediction models with a large state space (around 100 states) to be used in MPC.
without significantly impacting the run-time computational demand. What dominates the complexity of solving the QP is the number of optimisation degrees of freedom, which in the present work we keep low by using basis functions to map optimisation variables over the prediction horizon.

While the size of the state is not a computational concern, estimating the state accurately is a challenge, especially when there are many more states than measurements. The present work shows how linear feedback, in addition to the MPC optimal control action, can aid state estimation.

The ability to efficiently solve the QP despite a large state space allows us to use linear models directly from an aeroelastic design and simulation software such as Bladed. This brings benefits in industrial applications, since changes to the wind turbine model can be propagated into the linear model with minimal effort. Additionally, compared to first-principles modelling, aeroelastic linearisation captures more coupling between aerodynamics and the structure and requires less expertise to perform. Indeed, the key contribution of this paper is an MPC design process that accepts linear models with a large state space, can be easily configured and tuned and can be solved rapidly online.

Ground-breaking early work in the field was presented by Henriksen et al. A turbine model with drivetrain torsion, nacelle motion and actuator dynamics is linearised in four operating regions: minimum generator speed, variable generator speed, maximum generator speed below rated power, and above rated power. Offset-free control is achieved with disturbance modelling. State variables are estimated with one Kalman filter per region. Stability of each MPC controller is improved using a terminal set. The closed-loop behaviour is evaluated in a high-fidelity multi-body simulation.

While the modelling methods, application of MPC theory and validation in that work were thorough, switching between regions is not optimal, resulting in transients (albeit filtered) during region transitions. Periodic disturbances such as those caused by tower shadow and wind shear are mentioned by the authors but not considered in the MPC formulation. Such disturbances must be filtered out of the measurements so as to avoid unmodelled excitation of the controller. The present work shows that those filters must also be applied the prediction model to ensure reliable predictions.

Operating in multiple regions, with one linear model per region, was generalised further by Soliman et al. The algorithm in that work permits any number of regions, each of which contains a state estimator, the bank of which are simultaneously updated at all times. Transients between regions are reduced by calculating control action increments rather than absolute values. Using more numerous parallel models gives finer control over the operational objectives and reduces linearisation errors but comes with a computational cost. Robustness to model parameter uncertainty is investigated but the controller is not run in an aeroelastic simulation package.

Employing multiple linearisations improves optimality for a plant where the poles and zeros change in different operating regions, as they do in a wind turbine. However, they are not strictly required, if a decrease in performance is acceptable when operating away from a single linearisation point. The basic purpose of the controller is to maximise power capture subject to constraints on generator speed and torque and pitch actuator angle and velocity. Since constraints are handled naturally by MPC, the simplest approach with a single model is to define the cost function with a variable reference, which tracks the operating mode at run-time based on an estimate of the wind speed. Additionally, curtailment due to grid demands, wind farm control, noise modes or similar can be accommodated by reference tracking more straightforwardly than by multiple models. Quasi-LPV methods could be investigated in the future work to accommodate varying aerodynamics while maintaining a single QP to solve.

An alternative to reference tracking was presented by Gros and Schiltz and preceding work, called economic nonlinear MPC, where the cost function can directly maximise power capture using a nonlinear prediction model. This reduces the number of cost function parameters to tune but increases the computational complexity. Nevertheless, that work shows that such a problem can be solved in under 10 ms for a long prediction horizon on a fast computer. The present work poses the optimisation as a QP with reference tracking, but the algorithm could be adapted to use economic MPC in future work.

The industrial application of linear MPC for wind turbine control was considered by Laks, where the prediction model is generated from an aeroelastic code rather than from first principles. Some structural modes are disabled for the linearisation process, in order to give a model of appropriate complexity, yet drivetrain torsion and flapwise blade bending are included. Furthermore, blade root bending moments are included in the measurements, allowing individual pitch control (IPC). Disturbance accommodating control (DAC) and reference tracking allow for one model to operate across multiple regions. Simulations are performed with all structural degrees of freedom enabled.

A similar approach was taken by Guadayol, where the detailed linear model undergoes model reduction before being used to form state predictions. A technique for improving the linearisation to account for variations in rotor response with azimuth is presented, as is a way to interpolate between linear models and state estimators directly, rather than running them in parallel.

Model reduction comes with the penalty that the discarded modes could have a significant impact on closed-loop behaviour. The present work shows how underdamped structural modes can be controlled by incorporating linear feedback at run-time and into the prediction model, rather than by attempting to estimate all the necessary states online, which is hard to achieve accurately from the few measurements available. The same approach neatly addresses the need to filter periodic disturbances from the measurements without introducing additional model mismatch.

Due to time varying set-point changes, model-plant mismatch and stochastic disturbances, a terminal set cannot be meaningfully constructed. In the absence of a terminal set, a long prediction horizon can be helpful to improve closed-loop performance. The standard formulation of MPC has one degree of freedom per control action per time step in the prediction horizon. For example, a horizon of 10 s at 20 Hz, with two control actions would be 400 degrees of freedom, making the QP intractable in real time.
An alternative is to parameterise the control actions over a long prediction horizon as a weighted sum of exponential basis functions called Laguerre functions. As first introduced by Wang and Rossiter, this process greatly reduces the number of degrees of freedom in the QP. Indeed, the number does not depend on the sample rate or horizon length. Laguerre functions are orthonormal, exponentially decaying, and allow a trade-off between computational cost and conservativeness. Adegas et al apply Laguerre functions to wind turbine MPC, although the controller is not tested in full aeroelastic simulations, which the present work shows motivates additional design considerations.

In the work of Laks, Lio et al, Bottasso et al and others, light detection and ranging (LIDAR) measurements are incorporated to improve the turbine’s response to turbulence and gusts. MPC is not required in order to benefit from the preview information afforded by a forward-facing LIDAR system, since feed-forward control delivers this within the classical control framework. However, if the control actions approach their constraints in responding to preview information, then clearly MPC brings advantages over classical control. The same is true for IPC, that is, IPC is provided with classical control but MPC can do this while respecting limits on pitch angle, rate and acceleration. The present work does not include LIDAR or IPC, but our framework would facilitate such expansions in the future, using perhaps the modified constraints of Petrović et al.

By linearising directly from Bladed, incorporating linear feedback, measurement filters, reference tracking and defining the predicted future control actions in terms of Laguerre functions, the present work demonstrates how the three challenges of computational complexity, closed-loop performance and design complexity have been simultaneously solved in a practical way. Further practical considerations of the application of MPC in the real world include fault tolerance and cost function tuning. These and other works give confidence that MPC is now ready to be applied on a real turbine.

This paper is organised as follows. Section 2 takes the linear model as exported from Bladed and applies linear operations to produce a prediction model and Kalman filter. Section 3 parameterises predicted future control actions in terms of Laguerre functions, then provides a cost function and constraints, quadratic and linear in the Laguerre degrees of freedom. Section 4 demonstrates the offline design and online performance of the algorithm using a full fidelity Bladed model. Conclusions are given in Section 5.

## 2 PREDICTION MODEL

The turbine model is linearised at a single wind speed, giving the dynamical system matrices A, B, C, D and the steady-state inputs and outputs \( \bar{u}, \bar{y} \), which are related by Equation \( 1 \), where \( u_k, x_k, y_k \) are the input, state and measurement vectors at time step \( k \), respectively.

\[
\begin{align*}
    x_{k+1} &= Ax_k + Bu_k - u, \\
    y_k &= Cx_k + Du_k - \bar{u} + \bar{y}
\end{align*}
\] (1)

The linear model is obtained by linearising a high-fidelity multi-body turbine model, so it contains the aerodynamic coupling with the structures, actuator dynamics and more. The inputs are wind speed, pitch angle demand and generator torque demand. The outputs are measured generator speed and torque, pitch angle and rate, and nacelle fore-aft acceleration and velocity. For numerical conditioning, the linear model is scaled so that typical inputs and outputs are approximately in the range \( [-1, 1] \). This is achieved by pre-multiplying \( C \) and \( D \) by a diagonal output scaling matrix, and by post-multiplying \( B \) and \( D \) by a diagonal input scaling matrix. The result is a well-conditioned linear model, which we denote \( G_1 \).

We now apply a series of operations on the model, all of which maintain linearity. Bode diagrams illustrating these operations using a demonstration model are presented in Section 4.1.

The first operation ensures the model reflects the measurement filtering that must be applied at run-time, such as suppressing periodic disturbances at three times the rotor frequency (3P) or removing high-frequency noise. This is a practical way to prevent the closed-loop system from being excited by these unmodelled disturbances. Furthermore, measurement filters are used to assist with state estimation. While the Kalman filter could be used to estimate pitch rate from pitch angle and nacelle velocity from nacelle acceleration, it is more straightforward to achieve this with SISO filters. Pitch rate can be estimated with a differentiator and low-pass filter on pitch angle, and nacelle velocity can be estimated with an integrator and high-pass filter (to prevent drift) on nacelle acceleration. Denoting all such filters on the measurements as \( \mathcal{F} \), we can write the resulting linear model as

\[
    G_2 = \mathcal{F} G_1.
\] (2)

The second operation is to apply simple feedback from filtered measurements to control actions, which will be added to the output of the MPC at run-time. Under-damped structural modes are undesirable, because to be damped by the MPC, their states must be accurately estimated and the control actions being optimised must have sufficient degrees of freedom to dampen potentially many of these modes. For computational efficiency, we want to reduce the number of degrees of freedom in the QP.

Any number of linear transfer functions applied to the measurements is admissible in this step. Denoting the transfer functions that provide such feedback as \( \mathcal{C} \), we write the closed-loop model as

\[
    G_3 = G_2 (1 + G_2 \mathcal{C})^{-1}.
\] (3)
Thirdly, to accommodate wind speed variations, a disturbance model is connected to the model’s wind speed input (the first of three). Carefully designed transfer functions can be used for this,\textsuperscript{17} but for simplicity in this work, a first-order low-pass filter is used. The value of the time constant $\tau_v$ is not critical; we used 10 s. The gain $K_v$ is chosen so that if the input to the filter is unity variance white noise, the output is a signal with variance similar to normal turbulence. The resulting model can be written as

$$
G_v = G_v \begin{bmatrix} K_v/(\tau_v s + 1) & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.
$$

(4)

The model $G_v$ contains structural states, linear feedback states, measurement filter states and a wind speed state. All of these states are to be estimated by a Kalman filter, which takes filtered measurements and the previous time step’s optimal control action as inputs. It is a requirement of the Kalman filter being optimal that any noise on those measurements is white and that all inputs to the model are either known or are white noise. Augmenting the model with a wind model as in Equation (4) allows the Kalman filter to estimate some of the low frequency content of the turbulence (albeit with some lag) which the MPC can then reject.

3 | QP FORMULATION

From the separation principle,\textsuperscript{18} the estimated state at time step $k$, which we denote $x_{\text{est}}$, can be treated as the true state. All future states are dependent on this and the control action degrees of freedom. The predicted evolution of the state and outputs in the horizon is

$$
x_{i+1|k} = A_{m} x_{i|k} + B_{m} u_{i|k},
$$

$$
y_{i|k} = C_{m} x_{i|k} + D_{m} u_{i|k}.
$$

(5)

Here, $A_{m}, B_{m}, C_{m},$ and $D_{m}$ are the dynamical system matrices of $G_v$ and the subscript $i|k$ means $i$ time steps ahead of $k$. The number of time steps in the horizon is $N$, and the number of control actions per time step is $n_u = 2$, namely, pitch and generator torque. Predicted inputs $u_{i|k}$ and outputs $y_{i|k}$ are deviations from the linearisation point $\bar{u}, \bar{y}$ and are numerically conditioned by scaling as previously described.

In a standard formulation of linear MPC, each $u_{i|k}$ throughout the horizon would have one degree of freedom per control action, totalling $N n_u$. Instead, in this work we allow the engineer to choose the number $n_u$ of degrees of freedom per input, which does not depend on horizon length. The degrees of freedom, or optimisation variables, are arranged in a matrix $\Gamma \in \mathbb{R}^{n_u \times n_b}$. These are mapped onto the predicted inputs as follows:

$$
u_{i|k} = \Gamma \bar{\epsilon}_i, \quad \bar{\epsilon}_i = A_{\ell} \epsilon_{i-1}.
$$

(6)

Here, $A_{\ell}$ is a constant square real matrix, and $\bar{\epsilon}_0$ is a constant vector, both calculated offline as follows:

$$
A_{\ell} = \begin{bmatrix} a & 0 & 0 & 0 & \ldots \\ \beta & a & 0 & 0 & \ldots \\ -a\beta & \beta & a & 0 & \ldots \\ a^2 \beta & -a\beta & \beta & a & \ldots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \quad \epsilon_{0} = \sqrt{\beta} \begin{bmatrix} 1 \\ -a \\ a^2 \\ -a^2 \\ \vdots \end{bmatrix}.
$$

(7)

The real scalar $a \in (0, 1)$ is a time scaling parameter, and $\beta = 1 - a^2$. Each element of $\bar{\epsilon}_i$ as it evolves over the time horizon starting with $\bar{\epsilon}_0$ is one Laguerre function. The first function is a simple exponential decay, and each subsequent function has one more maximum or minimum before settling to zero. The time scaling should be set so that the maxima and minima of the Laguerre functions are well distributed in time throughout the horizon. Figure 1 shows the first nine Laguerre functions over the first 200 time steps.

With the dynamics in Equation (5), we can formulate the QP as follows:

$$
\min_{\Gamma} \sum_{i=1}^{N} (y_{i|k} - \bar{y}_k)^T Q (y_{i|k} - \bar{y}_k), \quad \text{s.t.} \begin{bmatrix} y_{i|k} \\ \bar{u}_{i|k} \end{bmatrix} \leq b, \quad i = 1, \ldots, N.
$$

(8)
Here, $\tilde{y}_k$ is the output reference at time step $k$, from a simple lookup table defined offline, with anemometer wind speed as the independent variable. Note that $\tilde{y}_k$ is calculated online at each time step, whereas $y$ is a constant defined offline during the linearisation process.

It is preferable to use an estimate of the rotor average wind speed than the nacelle anemometer measurement for the reference tracking. The Kalman filter provides such an estimate: it uses the whole rotor as an anemometer rather than a point on the nacelle. However, it is based on a linear model at a single operating point, so isn’t suitable for the whole operating range. A wind speed estimator could be used in future work.

The symmetric matrix $Q$ contains cost weights. This can be simply diagonal, making the number of parameters to tune very small. Since measured pitch angle and generator torque are outputs of the linear model in Equation (5), we do not need cost on the inputs, although that can be added if preferred.

The matrix $F$ and vector $b$ apply constraints on the predicted measurements and actions. We set $F$ to be zeros except for one element per row, which is either 1 or $-1$ depending on the orientation of the inequality required. Constraints on $y_{ik}$ apply to the outputs of the measurement filters $F$. Constraints imposed on $u_{ik}$ apply only to the actions optimised within the QP, not any additional action provided by the linear feedback $C$. If, on the other hand, the addition of any linear feedback control action is important to include in the constraints, and this can be achieved straightforwardly by adding an additional output to $\hat{y}$ that provides access to the control feedback states in the predicted outputs $y_{ik}$, then updating the relevant row of $F$ to include an extra 1 for that output.

We do not apply a terminal constraint, since in order to guarantee closed-loop stability, the terminal set and all constraints in the prediction horizon would need to be robust to all uncertainties in the system. Wind turbine design processes already stipulate a wide range of wind conditions that must be simulated with the controller in closed-loop, so it is reasonable to assume that if the controller is stable in those conditions then it will run successfully in the vast majority of real world conditions. Terminal set design for uncertain systems is an active research topic, and this could be reassessed in future work.

Likewise, there is no terminal cost in Equation (8). The impact on closed-loop behaviour is small provided the horizon is long. Again, a terminal cost can be included if preferred, and its cost matrix can be calculated using the tail of the Laguerre functions.

## 4 | DEMONSTRATION

### 4.1 | Modelling

We now demonstrate the above method with an example Bladed concept model. Selected steady-state inputs and outputs for each wind speed are given in Table 1. The steady-state outputs for pitch rate and nacelle velocity and acceleration are zero. This table is used online to calculate $\tilde{y}_k$.

The rated air gap power of the turbine is 2.1 MW, which results in a rated electrical power of 2.0 MW after losses. The gearbox ratio is 83, rotor diameter 80 m, and hub height 62 m. The Bladed model is linearised at 13 m/s. The linearised model initially contains 55 states: 30 blade states, 14 tower states, 5 drivetrain states and 6 pitch actuator states.

A notch filter is created for the measured generator speed. Its centre frequency is 1.89 rad/s, which is 3P (three times the rated rotor speed). The effect of this is shown in Figure 2. The same notch combined with a second-order low-pass filter is created for the nacelle acceleration.
measurement with cut-off frequency around the tower second modal frequency. Nacelle velocity is estimated from filtered nacelle acceleration using a first-order low-pass with cut-off frequency 1 rad/s. Blade pitch rate is estimated from measured pitch angle using a first-order high-pass filter with cut-off frequency 10 rad/s. Together, these filters form $\mathcal{F}$.

The only linear feedback $C$ we implemented was from filtered generator speed to torque demand. It provides edgewise blade damping and drivetrain damping and is implemented simply with a band-pass filter. The effect of this feedback is shown in Figure 3. More complex linear feedback is possible but was unnecessary for this demonstration. It is for the engineer to judge which way to provide modal damping is more successful out of MPC and SISO (or MIMO) design. After application of measurement filters, linear feedback and the disturbance model, the prediction model has 70 states.

For the Laguerre functions, we used $n_L = 9$, $\alpha = 0.9$ and $N = 200$, as shown in Figure 1. The sample rate is 20 Hz, so the horizon length is 10 s. The number of Laguerre functions was chosen by increasing it until the performance of the MPC in closed-loop stopped improving noticeably. The scaling parameter $\alpha$ was chosen so that the Laguerre functions all decay within the horizon, which was chosen to be long enough to cover several cycles of the first tower mode at 0.47 Hz. The cost matrix $Q$ is diagonal, with elements penalising selected predicted outputs: $Q_g$ for generator speed, $Q_p$ for pitch angle, $Q_t$ for generator torque and $Q_n$ for nacelle velocity.

The constraints are as follows: measured pitch angle between $-2$ degrees (fine pitch) and 45 degrees, measured pitch rate between $-5$ and $+5$ degrees per second and MPC demanded torque demand between zero and 13.4 kNm (rated torque). The resulting QP as per Equation (8) has 1200 element-wise inequalities and $n_n n_L = 18$ degrees of freedom. The solver we use is OSQP \(^{19}\).
4.2 | Simulations

The full aeroelastic turbine model is used in the following simulations, which are run in Bladed, with drag induction, tower shadow, dynamic inflow and dynamic stall all enabled. The controller, including filters, feedback and QP solver, is run in MATLAB, which communicates with Bladed every control time step via a dynamic link library (DLL).

4.2.1 | Steps

The first set of simulations consists of steps in hub wind speed from 6 to 19 m/s. The wind simulation is run multiple times with varying cost and constraints to highlight the impact of the tuning parameters on performance. Each step is an increase of 1 m/s over 1 s, and they are spaced 20 s apart. This is sufficient time for the dynamics to settle between each step. There is 8° flow inclination, 0.2 shear exponent, no turbulence and no yaw error.

Figure 4 shows the results of the first simulation of steps in wind speed, with default cost values. As the wind speed changes, the output reference changes according to the look-up table. Since the optimisation that the MPC algorithm performs aims to minimise the cost, which is quadratic in the difference between predicted and reference outputs, some behaviour in the variable speed range (before 100 s) can be seen that would not be seen in a classical wind turbine controller. That is, rather than the increased wind speed raising the generator speed, to which the torque loop responds by increasing torque demand, instead the torque demand drops substantially to raise the generator speed more quickly to the new reference output for generator speed. This behaviour is optimal with respect to the cost and, as we will see later, indicates the need for a more sophisticated output reference scheduling variable than simply using the anemometer wind speed.

Once the generator reaches rated speed of 157 rad/s, further steps in wind speed result in transients where the generator speed rises, the torque or pitch respond, sometimes with small overshoot, and everything settles rapidly. We now demonstrate how the tuning parameters affect the response of the turbine by repeating the simulation with small changes to Q or constraints.

Figure 5 shows a close-up of the steps at 40, 60, and 200 s. It is clear that the generator speed settles much faster and has less overshoot when the cost parameter on the generator speed is higher.

Figure 6 shows the pitch angle response at 200 and 220 s. In the left plot, we show that the pitch angle rises much more quickly and has more overshoot when the cost on generator speed is higher. In the right plot, we show that the pitch angle has a larger overshoot and faster settling time when the cost on pitch deviation is lower.

Figure 7 shows the steps from 80 to 130 s, comparing the responses of generator torque and speed to different costs on generator torque. When the cost is higher, the torque response is more damped, and there is more overshoot in speed, especially in variable speed operation.
Figure 8 shows two simulations with the lower pitch angle cost parameter, which increases the pitch activity, but in one case, the pitch rate limit is set far away at 5 deg/s, effectively unconstrained, and the other it is set very tightly at 1 deg/s. The constraint causes the pitch angle to take longer to rise, which means it has to rise higher to stabilise the generator speed. That explains the corrections that follow each step, where the pitch rate goes substantially more negative. At higher wind speeds, for example, 260 s onward, the generator speed is more sensitive to pitch angle changes, so the pitch rate constraint is not hit, and so the negative pitch rate correction is not seen.

4.2.2 | Turbulence

We now run a 300 s simulation with turbulent wind. The mean wind speed is 14 m/s, the longitudinal turbulence intensity is 16%, lateral 13% and vertical 8%. Again the flow inclination is 8°, shear exponent 0.2 and no mean yaw misalignment. Figure 9 shows the response of the generator speed, pitch angle, generator torque and nacelle displacement. The following observations on the results give some insights into the working of the control algorithm.

Due to the relatively high wind speed, the pitch is active for the entire simulation, only hitting fine pitch briefly, yet the torque demand is also quite active. This is because there is no constraint on the torque being held at rated while the pitch is active like there is in a classical wind turbine control algorithm. The internal prediction model contains pitch system dynamics, so the optimiser will use drops in torque demand to provide
Figure 6. (Left) Pitch angle response with \( Q_g = 1.0 \) (blue) and \( Q_g = 0.1 \) (red). (Right) Pitch angle response with \( Q_p = 0.3 \) (blue), \( Q_p = 1.0 \) (red) and \( Q_p = 2.0 \) (green).

Figure 7. Generator torque (left) and generator speed (right) responses with \( Q_t = 1.0 \) (blue) and \( Q_t = 3.0 \) (red).

Figure 8. Pitch rate response with the pitch rate constraint set to 5 deg/s (blue) and to 1 deg/s (red). The cost \( Q_p = 0.3 \) for both simulations.
short-term increases to generator speed. Furthermore, sudden changes to the pitch angle result in higher nacelle velocities, which are penalised in the cost function. It is a straightforward process to alter the trade-off between these effects, by tuning the cost matrix $Q$.

At around 65 s and again at around 160 s, the pitch angle rises despite the generator speed being close to 100% of nominal. This causes the generator speed to drop, causing a drop in torque some seconds later that seems undesirable. This behaviour might be something the engineer could look for when tuning the cost matrix. It is caused by the output reference $\tilde{y}_k$ shifting, which is based on the anemometer measurement of the wind speed at the nacelle. Due to spatial variation in the wind across the rotor, the anemometer measurement will sometimes not be a good indicator of the rotor average wind speed, which is what passes through the rotor plane and causes aerodynamic thrust and torque. Future work could use wind speed estimators for this purpose.

The nacelle displacement time series in Figure 9 shows quasi-steady-state variations that correspond to variations in thrust from wind speed and pitch angle, and additional frequency content from structural modes. It is clearer to demonstrate the effectiveness of the controller at reducing fore-aft motion by comparing the power spectral density (PSD) of simulations with different values of $Q_n$, the cost on nacelle velocity.

Figure 10 shows that when the cost function does not penalise nacelle velocity, there is a substantial peak at the tower first fore-aft mode's natural frequency of 0.47 Hz.

4.3 Performance of Laguerre parameterisation

While the implementation of the QP solver is not the focus of this work, we now present some results regarding the run-time performance of the presented MPC controller in the turbulence simulation. The performance is compared to the same simulation, with the same QP solver, cost and constraints, but without the Laguerre parameterisation of the degrees of freedom, that is, a standard MPC configuration with one optimisation
variable per control action, per prediction step. In the Laguerre case, the average time taken to solve the QP was 9 ms. In the standard MPC case, it was 95 ms. The processor is one core of an Intel i5.

The Laguerre parameterisation is a trade-off whereby computational cost is reduced, but the solution is less optimal due to the reduced degrees of freedom in the QP. To quantify this, the cost function was evaluated for the measured signals over the course of the turbulent simulation, again comparing Laguerre with standard MPC. The cost was 1.7% higher in the Laguerre case with $n_L = 9$. The simulation was rerun with $n_L = 20$ and the cost reduced to 1.1% higher than under the standard configuration.

5 CONCLUSIONS

When applying MPC to wind turbine control, three challenges arise: computational complexity, model-plant mismatch and design complexity. This paper shows how to solve all three challenges and demonstrates the algorithm by running deterministic and turbulent aeroelastic simulations in Bladed with full realism enabled.

Computational complexity is tackled by parameterising the predicted control actions with exponential basis functions, specifically Laguerre functions. The control actions are defined over a long prediction horizon in terms of only a few optimisation variables. Since the degrees of freedom are thereby too few to target specific underdamped modes such as blade edgewise vibration, the MPC control action is supplemented with user-defined linear feedback loops where necessary. The cost and constraints in the QP can account for the contributions of both the Laguerre actions and the linear feedback. The Kalman filter that estimates the states for the MPC predictions no longer needs to be so carefully designed to correctly estimate such hard to observe modes.

Model-plant mismatch is minimised by taking a linear model, with as many states as desired, directly from the aeroelastic code. This makes it as similar as possible to the aeroelastic model. Even so, some dynamics are omitted by the linearisation process, such as periodic aerodynamic disturbances. The solution presented in this work is to filter the measurements at run-time and apply the same transfer functions to the linear model at design-time to ensure their impact is accounted for in the predictions. Simulations show good robustness across a wide range of wind speeds, although additional linearisations would help towards the top of the operating range.

Taking the linear model directly from the aeroelastic code also brings benefits in reducing design complexity. In an industrial wind turbine design context, models are frequently changed, requiring the controller to be re-tuned each time. In our framework, once the filters and any user-defined feedback have been defined, no additional tuning is required, even after changes to the underlying turbine model. The framework requires only trivial parameters, namely, four cost values and any applicable constraint values, then all interactions between pitch and torque are accounted for directly in the QP. Rotor speed references are defined in a look-up table, which can easily be extended to accommodate power curtailment demands, for example, from the grid or wind farm control.

The MPC scheme presented in this paper successfully operates the turbine above and below rated wind speed, naturally handling the transitions from variable to rated speed and from partial to full power. The framework is ready to accommodate future extensions such as multiple linearisations, IPC, LIDAR and variable power set-points.

Figure 10  Power spectral density of nacelle fore-aft velocity for a turbulence simulation with mean wind speed 14 m/s and with varied cost on nacelle velocity: $Q_n = 0$ (blue) and $Q_n = 1$ (red)
ACKNOWLEDGEMENT
This research was supported by the European Union's Horizon 2020 research and innovation programme under Grant 727680 (TotalControl, https://www.totalcontrolproject.eu).

PEER REVIEW
The peer review history for this article is available at https://publons.com/publon/10.1002/we.2695.

DATA AVAILABILITY STATEMENT
The data that support the findings of this study are available from the authors upon reasonable request.

ORCID
Martin A. Evans https://orcid.org/0000-0002-4287-6313
Wai Hou Lio https://orcid.org/0000-0002-3946-8431

REFERENCES

How to cite this article: Evans MA, Lio WH. Computationally efficient model predictive control of complex wind turbine models. Wind Energy. 2022;25(4):735-746. doi:10.1002/we.2695