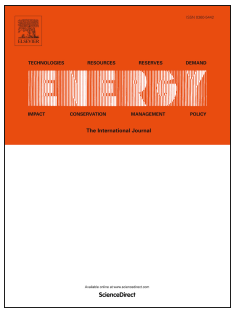
Accepted Manuscript



Modelling and control of a wind turbine and farm

S. Hur

PII: S0360-5442(18)30891-0

DOI: 10.1016/j.energy.2018.05.071

Reference: EGY 12906

To appear in: *Energy*

Received Date: 9 February 2018

Revised Date: 26 April 2018

Accepted Date: 9 May 2018

Please cite this article as: Hur S, Modelling and control of a wind turbine and farm, *Energy* (2018), doi: 10.1016/j.energy.2018.05.071.

Modelling and Control of a Wind Turbine and Farm

S. Hur[a,*]

[a]*School of Electronics Engineering, Kyungpook National University, Daegu, South Korea*

Abstract

The Matlab/Simulink model of the Supergen (Sustainable Power Generation and Supply) Wind 5 MW exemplar wind turbine, which has been employed by a number of researchers at various institutions and Universities over the last decade, is reported. It is subsequently improved, especially in speed, to facilitate wind farm modelling, which usually involves duplicating wind turbine models. The improvement is achieved through various stages, including prewarping, discretisation using Heun's method in addition to Euler method, and conversion to C. Results are presented to demonstrate that improvement in speed is significant and that the resulting wind turbine model can be used for wind farm modelling more efficiently. It is important to highlight that improvement in speed is achieved without compromising the complexity of the turbine model; that is, each turbine included in a wind farm is neither simplified nor compromised. The use of the wind farm model for testing a wind farm controller that has recently been introduced is also demonstrated.

Keywords:
Wind farm modelling, wind turbine modelling, wind farm control

1. Introduction

The 2016 statistic publication by WindEurope reports that there are 153.7 GW of installed wind energy capacity in the European Union. With such high penetration of wind power, the power generated by wind farms can no longer simply be that dictated by the wind speed. The power output of some wind turbines is already being curtailed [1, 2]. It will be necessary for wind farms to provide services to the grid including spinning reserve, frequency support and assistance with supply-demand matching, which could be achieved by the use of an appropriate wind farm controller. In order to design a wind farm controller,

[☆]This research was supported by Kyungpook National University Research Fund, 2017. This work was also supported by the EPSRC EP/N006224/1 "Maximising wind farm aerodynamic resource via advanced modelling (MAXFARM)" and FP-ENERGY-2013-IRP GAN-609795 "Integrated Programme on Wind Energy (IRPWind)".

[*]Corresponding author
Email address: hur.s.h@ieee.org (S. Hur)

an efficient wind farm model, which can be executed fast and is detailed at the same time, needs to be developed – this is the main objective of the work presented in this paper.

In more detail, the Matlab/Simulink® (Matlab) model of the Supergen (Sustainable Power Generation and Supply) Wind 5 MW exemplar wind turbine includes modules of aerodynamics, blades dynamics, rotor dynamics, actuator dynamics, drive-train, tower dynamics, generator, etc. The 5 MW wind turbine model is selected because it is the largest wind turbine model available within the consortium. Currently, Matlab and Bladed models of the 8 MW exemplar wind turbine is being developed [3], and the work presented here will be applied to the 8 MW model once it becomes complete and available. The 5MW model was first introduced in 2000 [4], and has since been updated/improved and carefully validated/tuned using the high fidelity aero-elastic model, i.e. in DNV-GL Bladed (Bladed), of the same exemplar turbine. It is important to point out that in the wind energy sector, validation of a wind turbine model using aeroelastic software, such as Bladed and FAST by the National Renewable Energy Laboratory (NREL), is a common practice and is widely accepted in both industry and academia because it is not convenient and often infeasible to utilise real-life turbines due to their size and availability. As an example, in industry, wind turbine controllers are applied to and tuned using an aeroelastic wind turbine model before it is applied to the real-life wind turbine [5, 6]. The validated wind turbine model has been utilised for various projects [7, 8] over the last decade, especially within the Supergen Wind Consortium.

Over the last few years, the model has also been utilised for developing wind farm models [9, 10, 11]. The wind farm models require the wind turbine model to be duplicated and, in turn, the simulation time could be increased exponentially – this is when the full dynamics of each turbine model is retained and, therefore, each model is not simplified or compromised.

To develop a wind farm model including as many as hundreds of the turbine models (each turbine without simplification), we propose to discretise the turbine model and convert it to C code. In more detail, the drive-train module which constitutes the wind turbine model is stiff. The numerical integration performed by the Simulink solver during simulation is not optimal especially when the model is stiff, but the non-optimal numerical integration can be prevented by optimally discretising the model in advance, thereby avoiding the numerical integration completely.

Stiff systems require a more stable "implicit" discretisation method, hence Heun's method is utilised, which is further improved. Moreover, stiff systems are known to potentially become unstable when discretised, which should be taken into account when discretising. This phenomenon is avoided by "prewarping" [12] in advance. Once the model has been optimally discretised (to improve the simulation speed), it is converted to C for further improvement. In other words, the model is improved in simulation speed through two stages:

1. manual, optimal discretisation to prevent the Simulink solver from non-optimally performing numerical integration for simulating the stiff drive-

train module

2. conversion of the discretised model to C.

Note that discretisation is required not only to avoid non-optimal numerical integration, but also to facilitate conversion of the model to C; that is, without discretisation, conversion of the model to C cannot be conducted.

Simulation results are presented to demonstrate that improvement in simulation speed due to the optimal discretisation and conversion to C is promising, which makes the model more suitable for wind farm modelling.

An efficient wind farm controller is reported in [11]. Then, the wind farm controller could only be tested by application to a wind farm model of 10 turbines. The wind farm model developed in this paper allows the wind farm controller to be tested against a larger wind farm model of 30 turbines. Simulation results are also presented to demonstrate the wind farm controller is scalable; that is, it can be applied to a larger wind farm model.

The work presented here thus focuses on modelling, simulation and control application, rather than the introduction of a new controller design. It will serve as the basis for the development and analysis of new and existing wind farm control designs. More precisely, the first contribution of this work is the development of a more efficient wind farm model through the use of Heun's method, Euler's method, prewarping, etc. Each turbine model included in the wind farm model is based on the existing Supergen model, but the wind farm model is significantly faster without compromising the complexity of each turbine model. The second contribution is that the wind farm controller that has recently been introduced and tested against a wind farm model of 10 turbines is applied to another wind farm model that is significantly larger, thereby being more realistic, i.e. a significant increase in the wind farm size is only feasible as a result of the first contribution.

In Section 2, the wind turbine model is reported. In Section 3, the wind turbine model is discretised and converted to C. In Section 4, the discretised wind turbine model in C is extended to a wind farm model, and simulation results of the wind farm model are presented. The use of the wind farm model for testing a wind farm controller is demonstrated in Section 5, and conclusions are drawn in Section 6.

2. Wind Turbine Model

The parameters and variables used for the wind turbine model reported in this section are first summarised as follows:

B_T	tower fore-aft damping moment [Nm]
C_o	out-of-plane bending moment coefficient
C_p	aerodynamic power coefficient
C_T	thrust coefficient
F_t	thrust force [N]
h	tower height [m]
I_1	inertia of the low-speed shaft [kg m^2]
I_2	inertia of the high-speed shaft [kg m^2]
J	rotor inertia [kg m^2]
J_C	tower/rotor cross-coupling inertia [kg m^2]
J_T	tower fore-aft inertia [kg m^2]
K_a	tower shape factor
K_T	tower fore-aft stiffness [Nm/rad]
K_1	low-speed shaft stiffness [Nm/rad]
K_2	high-speed shaft stiffness [Nm/rad]
$M_{A\theta_R}$	in-plane rotor torque [Nm]
$M_{A\phi_R}$	out-of-plane rotor torque [Nm]
N	gearbox ratio
R	rotor radius [m]
T_{Gen}	generator torque
T_H	hub torque [Nm]
V	wind speed [m/s]
β	pitch angle [rad]
η	drive-train damping ratio
γ^*	drive-train material damping
λ	tip speed ratio
Ω	in-plane rotor speed [rad/s]
Ω_R	rated in-plane rotor speed [rad/s]
ω_e	rotating blade edge natural frequency [rad/s]
ω_{es}	stationary blade edge natural frequency [rad/s]
ω_f	rotating blade flap natural frequency [rad/s]
ω_{fs}	stationary blade flap natural frequency [rad/s]
ϕ_H	out-of-plane displacement of hub [rad]
ϕ_R	out-of-plane displacement of rotor [rad]
ρ	density [kg/m^3]
θ_H	angular displacement of hub [rad]
θ_R	angular displacement of rotor [rad]

The model can be classified into pitch module, aerodynamic module, rotor

module, drive-train and tower module, etc.

2.1. Pitch Module

The pitch mechanism is modelled using the following second order transfer function:

$$P(s) = \frac{c_1}{s^2 + c_2 s + c_1} \tag{1}$$

where c_1 are c_2 are parameters defined by Supergen Wind. The input and output are respectively pitch angle demand and actual pitch angle.

2.2. Aerodynamics module

In-plane rotor torque is given by

$$M_{A\theta_R} = \frac{1}{2}\rho\pi V^2 R^3 \frac{C_p(\lambda,\beta)}{\lambda} \tag{2}$$

where the tip-speed ratio, λ, is defined as

$$\lambda = \frac{R\Omega}{V} \tag{3}$$

Out-of-plane rotor torque is given by the following equation

$$M_{A\phi_R} = \frac{1}{2}\rho\pi V^2 R^3 C_o(\lambda,\beta) \tag{4}$$

Thrust force is given by

$$F_t = \frac{1}{2}\rho\pi V^2 R^2 C_T(\lambda,\beta) \tag{5}$$

The aerodynamic power coefficient, C_p, out-of-plane bending moment coefficient, C_o, and thrust coefficient C_T are obtained using Bladed.

It is important to point out that the aerodynamic module presented here is in the simplest form. However, aerodynamic effects of wind turbines in a wind farm can significantly limit the amount of power that can be extracted from a given wind farm footprint. Therefore, the module will be improved in the near future to include more details and complexity to take into account these effects, similarly to the aerodynamic module introduced in [13]. Moreover, it will also be improved to incorporate the effect of having "upwind" wind turbines, concerned in this paper, in more detail.

2.3. Rotor module

The Lagrangian for the rotor dynamics is [14]:

$$\begin{aligned}L_R = &0.5(J\dot{\theta}_R^2 + J\dot{\phi}_R^2 + J_T\dot{\phi}_T^2 + J_C\dot{\phi}_R\dot{\phi}_T)\\ &- 0.5K_e[(\theta_R-\theta_H)cos\beta - (\phi_R - K_a\phi_T)sin\beta]^2\\ &- 0.5K_f[(\theta_R-\theta_H)sin\beta + (\phi_R - K_a\phi_T)cos\beta]^2\\ &- 0.5J(\omega_e^2-\omega_{es}^2)(\Omega^2/\Omega_R^2)(\theta_R-\theta_H)^2\\ &- 0.5J(\omega_f^2-\omega_{fs}^2)(\Omega^2/\Omega_R^2)(\phi_R-K_a\phi_T)^2 - 0.5K_T\phi_T^2 + D_T\phi_T\\ &+ M_\theta(\theta_R-\theta_H) + M_\phi(\phi_R-K_a\phi_T) + F_Th\phi_T \end{aligned} \tag{6}$$

The rotor dynamics is modelled by the following equations of motion:

$$\begin{aligned}\ddot{\theta}_R = &-\omega_{es}^2[(\theta_R-\theta_H)cos\beta - (\phi_R-K_a\phi_T)sin\beta]cos\beta\\ &-\omega_{fs}^2[(\theta_R-\theta_H)sin\beta + (\phi_R-K_a\phi_T)cos\beta]sin\beta\\ &-(\omega_e^2-\omega_{es}^2)(\Omega^2/\Omega_R^2)(\theta_R-\theta_H) + M_\theta/J\end{aligned} \tag{7}$$

$$\begin{aligned}\frac{JJ_T-J_C^2}{JJ_T+K_aJJ_C}\ddot{\phi}_R = &\omega_{es}^2[(\theta_R-\theta_H)cos\beta - (\phi_R-K_a\phi_T)sin\beta]sin\beta\\ &-\omega_{fs}^2[(\theta_R-\theta_H)sin\beta + (\phi_R-K_a\phi_T)cos\beta]cos\beta\\ &-(\omega_f^2-\omega_{fs}^2)(\Omega^2/\Omega_R^2)(\theta_R-K_a\theta_T) + M_\phi/J\\ &+\frac{J_C/J}{J_T+K_aJ_C}(K_T\phi_T + B_T\dot{\phi}_T - hF_T)\end{aligned} \tag{8}$$

$$\begin{aligned}\frac{JJ_T-J_C^2}{JJ_T+K_aJ^2}\ddot{\phi}_T = &-\omega_{es}^2[(\theta_R-\theta_H)cos\beta - (\phi_R-K_a\phi_T)sin\beta]sin\beta\\ &+\omega_{fs}^2[(\theta_R-\theta_H)sin\beta + (\phi_R-K_a\phi_T)cos\beta]cos\beta\\ &+(\omega_f^2-\omega_{fs}^2)(\Omega^2/\Omega_R^2)(\theta_R-K_a\theta_T) - M_\phi/J\\ &-\frac{1}{J_C+K_aJ}(K_T\phi_T + B_T\dot{\phi}_T - hF_T)\end{aligned} \tag{9}$$

The hub torque dynamics is described by the following equation:

$$\begin{aligned}T_H = &J\omega_{es}^2[(\theta_R-\theta_H)cos\beta - (\phi_R-K_a\phi_T)sin\beta]cos\beta\\ &+J\omega_{fs}^2[(\theta_R-\theta_H)sin\beta + (\phi_R-K_a\phi_T)cos\beta]sin\beta\\ &-J(\omega_e^2-\omega_{es}^2)(\Omega^2/\Omega_R^2)(\theta_R-\theta_H)\end{aligned} \tag{10}$$

Combining Equations (7), (8), (9), and (10), a 6th order nonlinear model is obtained.

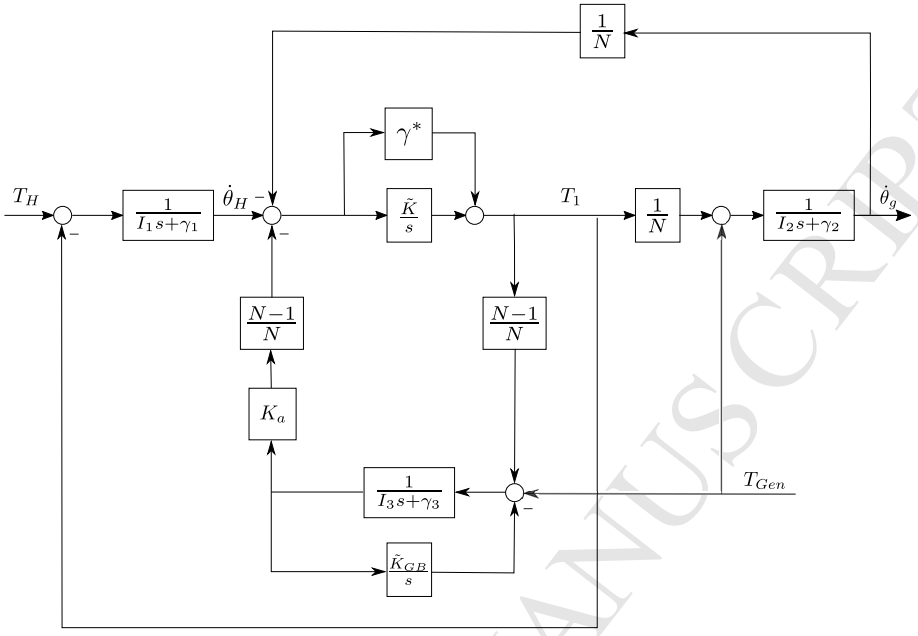


Figure 1: Drive-train module.

2.4. Drive-train and tower module

The drive-train module [15] is depicted in Figure 1. γ^* in the figure denotes drive-train material damping, a function of K_1, K_2, N, η, I_1, and I_2, i.e.

$$\gamma^* = 2\eta\sqrt{\tilde{K}\tilde{I}}\frac{(K_1 + N^2K_2)^2}{(N^2K_2)^2} \tag{11}$$

where

$$\tilde{K} = \frac{N^2K_1K_2}{K_1 + N^2K_2} \tag{12}$$

and

$$\tilde{I} = \frac{N^2I_1I_2}{I_1 + N^2I_2} \tag{13}$$

The fifth order linear model is stiff, causing the Simulink numerical solver to choose a non-optimally large sampling step during simulation, thereby increasing the simulation speed. As previously mentioned, the model is optimised in terms of speed via discretisation and conversion to C as reported in Section 3.

2.5. Other dynamics and the turbine controller

Further dynamics and modules included in the model, but not presented in this paper include wind speed model [16], rotational sampling (i.e. 1P, 2P and

3P), and unsteady aerodynamics [10]. If necessary, the model will be improved in future to include further details and complexity, such as more detailed effects of material properties on aeroelastic behaviour of wind turbines, examples of which can be found in [17, 18].

The commercially used Supergen Wind turbine controller for this turbine model is reported in [19] and is discretised in [9]. The discretised controller is combined with the discretised turbine model in Section 3 for simulations throughout this paper.

2.6. Validation

As previously mentioned, this model has been used at various institutions and universities and has been validated thoroughly by various researchers. The most recent validation results using Bladed can be found in [20].

3. Discretisation and Conversion to C

3.1. Discretisation

Apart from the drive-train module, the model is non-stiff and nonlinear, and the Forward Euler method can thus be used to discretise all the modules except the drive-train module. Note that nonlinearity makes it difficult and impractical to utilise implicit discretisation methods, such as the Backward Euler or Heun's method. However, because the modules are non-stiff, the use of "explicit" Forward Euler method instead of implicit methods is sufficient in any case.

The drive-train module, on the other hand, is stiff and linear. Stiff systems have poles of very different magnitudes, i.e. some poles close to the origin (in the s-plane) and some poles very far away from them, meaning that they contain both slow and fast dynamics. This, therefore, makes it difficult for the numerical integration solver of Matlab/SIMULNIK to choose a correct and optimal sampling step when simulating the stiff drive-train module, and the solver thus has a strong tendency to opt for a very small sampling step, as opposed to a larger, more optimal sampling step, resulting in a significant increase in simulation time.This phenomenon is avoided here by optimally discretising the stiff drive-train module in advance, thereby not requiring numerical integration to be carried out at all during simulation.

Stiff systems require an implicit discretisation method, which is more stable than explicit discretisation methods. Heun's method is thus chosen to discretise the drive-train module. Fortunately, the model being linear makes it simpler for an implicit discretisation method to be applied. Further, stiff systems have some of their poles close to the origin in the s-plane, which tend to end up outside the unit circle when discretised, leading to instability. This phenomenon is prevented here by prewarping. Prewarping involves moving the poles of the original continuous drive-train module slightly further away from the origin in the s-plane such that the poles of the discretised drive-train module still remain

within the unit circle in the z-plane when discretised. This is achieved here by increasing the drive-train material damping γ^* of the drive-train module.

Since the drive-train module is linear, it can readily be represented in the following standard continuous state space form:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + B\mathbf{u}(t)\\ \mathbf{y}(t) &= C\mathbf{x}(t) + D\mathbf{u}(t)\end{aligned} \tag{14}$$

where the inputs $\mathbf{u} \in \mathbb{R}^2$ denote aerodynamic torque and torque demand, the outputs $\mathbf{y} \in \mathbb{R}^2$ aerodynamic speed and generator speed, and $\mathbf{x}(t) \in \mathbb{R}^5$ the states.

Since one of the inputs is continuous and the other discrete, Equation (14) is split into two equations: one including hub torque, $u_1(t)$, and the other including torque demand, $u_2(t)$. Subsequently, the two equations are discretised differently. By applying Heun's method to the first one as follows

$$\mathbf{x}(n+1) = \mathbf{x}(n) + \frac{T_s}{2}\left[(A\mathbf{x}(n+1) + Bu(n+1) + A\mathbf{x}(n) + Bu_1(n))\right] \tag{15}$$

the following discretised equation is obtained:

$$\mathbf{x}(n+1) = \underbrace{(I - \frac{T_s}{2}A)^{-1}(I + \frac{T_s}{2}A)}_{E}\,\mathbf{x}(n) + \underbrace{(I - \frac{T_s}{2}A)^{-1}\frac{T_s}{2}B}_{F}\,u_1(n+1) \tag{16}$$

$$\mathbf{y}(n) = C\mathbf{x}(n) \tag{17}$$

This equation is not in the standard state space form due to the implicit term $Fu_1(n+1)$. This term is, in effect, equivalent to the direct-feed term and hence can be moved to Equation (17) such that

$$\mathbf{y}(n) = C\bar{\mathbf{x}}(n) + CFu_1(n) \tag{18}$$

Note that because the equation has been modified, the states must also be modified from $\mathbf{x}(n)$ to $\bar{\mathbf{x}}(n)$. Taking this into account, the following discretised model can be derived.

$$\begin{aligned}\bar{\mathbf{x}}(n+1) &= E\bar{\mathbf{x}}(n) + (EF + F)u_1(n)\\ \mathbf{y}(n) &= C\bar{\mathbf{x}}(n) + (CF + D)u_1(n)\end{aligned} \tag{19}$$

Heun's method essentially accepts the mean of $\mathbf{u}(n)$ and $\mathbf{u}(n+1)$, i.e. $(\mathbf{u}(n)+\mathbf{u}(n+1))/2$. This is suitable for $u_1(t)$, which comes from another continuous module, aerodynamic module. The other input, $u_2(t)$, is received from the discrete controller, and the drive-train module needs to take $u_2(n)$ over the sample interval instead of the mean. Therefore, Equation (16) is modified as follows for the second input $u_2(n)$.

$$\begin{aligned}\mathbf{x}(n+1) &= E\mathbf{x}(n) + 2Fu_2(n)\\ \mathbf{y}(n) &= C\mathbf{x}(n) + Du_2(n)\end{aligned} \tag{20}$$

In summary, when the input is aerodynamic torque, which is originally continuous, we use Equation (19), and when the input is generator torque demand, which is discrete from the discrete controller, we use Equation (20). These two models can be combined to obtain one state space equation that accepts both the inputs, hence the model becomes multi-input multi-output (MIMO) as follows.

$$\begin{aligned} \mathbf{x}(n+1) &= E\mathbf{x}(n) + G\mathbf{u}(n) \\ \mathbf{y}(n) &= C\mathbf{x}(n) + H\mathbf{u}(n) \end{aligned} \tag{21}$$

where $\mathbf{u} \in \mathbb{R}^2$, $\mathbf{y} \in \mathbb{R}^2$, and $\mathbf{x} \in \mathbb{R}^5$ denote the same inputs, outputs and states as the original continuous module in Equation (14). G and H are given as follows:

$$G = \begin{bmatrix} EF_1 + F_1 & 2F_2 \end{bmatrix} \tag{22}$$

$$F = \begin{bmatrix} F_1 & F_2 \end{bmatrix} \tag{23}$$

$$H = \begin{bmatrix} J_1 & 0 \\ J_2 & 0 \end{bmatrix} \tag{24}$$

$$J = \begin{bmatrix} J_1 & J_2 \end{bmatrix}^T = CF_1 + D_1 \tag{25}$$

$$D = \begin{bmatrix} D_1 & D_2 \end{bmatrix} \tag{26}$$

Equation (21) now accepts both aerodynamic torque and generator torque demand producing the outputs of aerodynamic speed and generator speed.

As previously mentioned, the rest modules are non-stiff and nonlinear, so the Forward Euler method is used to discretise them. Note that nonlinearity makes it difficult and impractical to utilise implicit discretisation methods, such as Backward Euler or Heun's method. However, as the modules are non-stiff, the use of an explicit Forward Euler method is sufficient anyway. The discretised model of the rest dynamics is combined with that of the drive-train module resulting in the following combined state space model ready for conversion to C:

$$\mathbf{x}(n+1) = A\mathbf{x}(n) + B\mathbf{u}(n) \tag{27}$$

$$\mathbf{y}(n) = C\mathbf{x}(n) + D\mathbf{u}(n) \tag{28}$$

where $\mathbf{u}(n) \in \mathbb{R}^3$ are torque demand, pitch demand and wind speed, and $\mathbf{y}(n) \in \mathbb{R}^7$ include hub torque, generator speed, hub speed, generator speed, nacelle acceleration, etc. The size of $\mathbf{y}(n)$ can readily be increased to include any states $\mathbf{x}(n) \in \mathbb{R}^{15}$.

3.2. Prewarping

Drive-train material damping γ^* in Equation (11) is adjusted prior to discretisation to prevent the discretised model from becoming unstable and to ensure that the discretised module follows the characteristics of the original

continuous module more closely. This process is known as prewarping. In more detail, γ^* in Equation (11) is adjusted and the change is subsequently incorporated into the module such that the frequency response of the drive-train module is modified to yield an improved frequency response as follows. Figure 2 depicts the open-loop frequency responses of the drive-train module. The blue plot is the response of the original continuous model (from Section 2), and the red plot is the response of the discretised model as a result of the discretisation described above; that is, without prewarping. The yellow plot is the response of the discretised model that is prewarped and, in turn, discretised. The responses illustrate that prewarping prior to discretisation allows the discretised model to exhibit more similar characteristics to the original continuous model than the discretised model that is not prewarped in advance. More importantly, without prewarping the increased peak at around 13.7 rad/s shown in Figure 2 causes the discretised module to become eventually unstable.

3.3. Conversion to C

The continuous model must be discretised before it can be converted to C, and this is one of the two reasons that the model is discretised in Section 3.1 – the other reason is to avoid non-optimal numerical integration during simulation as previously mentioned. Following the discretisation process described above, the model can now be converted to C as follows:

1. The continuous model, including various modules, described in Section 1 is discretised, using the Forward Euler and Heun's methods.
2. The discretised model is written in an s-function; that is, the entire model is written in one s-function incorporating all the parameters, inputs and outputs defined in Section 2.
3. The discrete s-function is rewritten in C.
4. The C file is converted to C MEX for simulation in Matlab/SIMULINK.

As a result, the Simulink model is much more efficient and fast as it utilises C functions instead during simulation. The parameters are loaded in the Matlab workspace before simulation. Importantly, the discretised model provides the same complexity and details as the original continuous model; that is, no dynamics is excluded as a result of the discretisation and conversion to C although improvement in simulation speed is significant as reported in Section 4.

4. Wind Farm Model

4.1. Preliminary Description of Results

The discretised (prewarped in advance) model is simulated in Matlab/SIMULINK in comparison to the original continuous model for 600 seconds for wind farms of 1, 10 and 100 turbines on an Intel® Core™i7-6700 3.40GHz (8 CPU) machine. The results are shown in Table 1. Note that each turbine includes the same full envelope controller that is discretised in [9], but no wind farm controller is included at this stage.

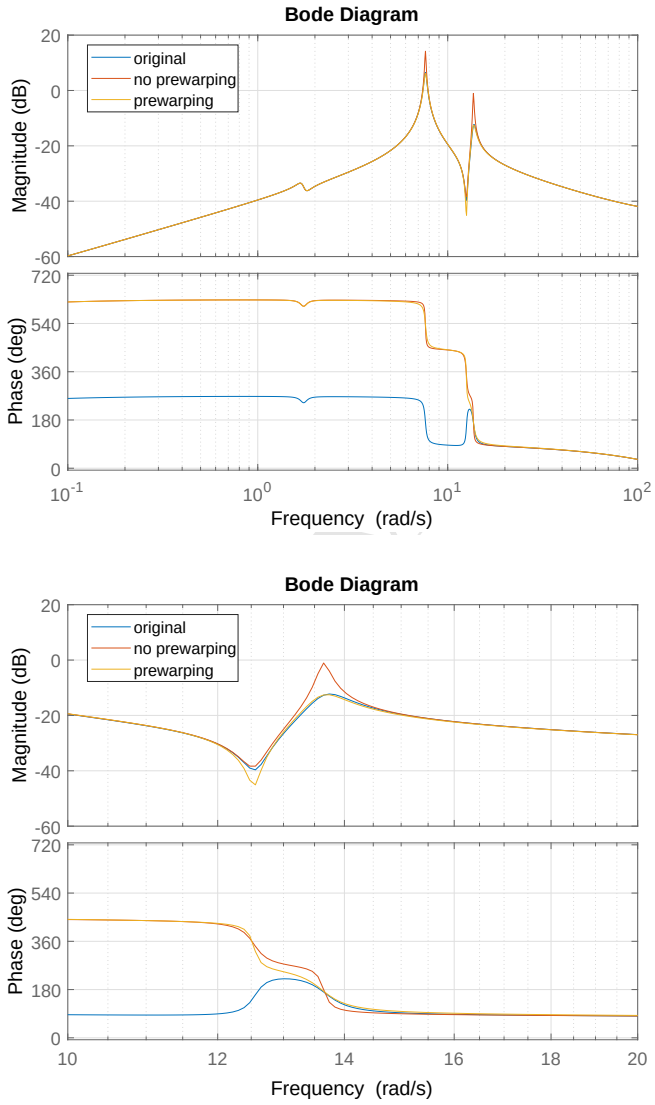


Figure 2: Discretisation and prewarping; the second one shows a zoomed version.

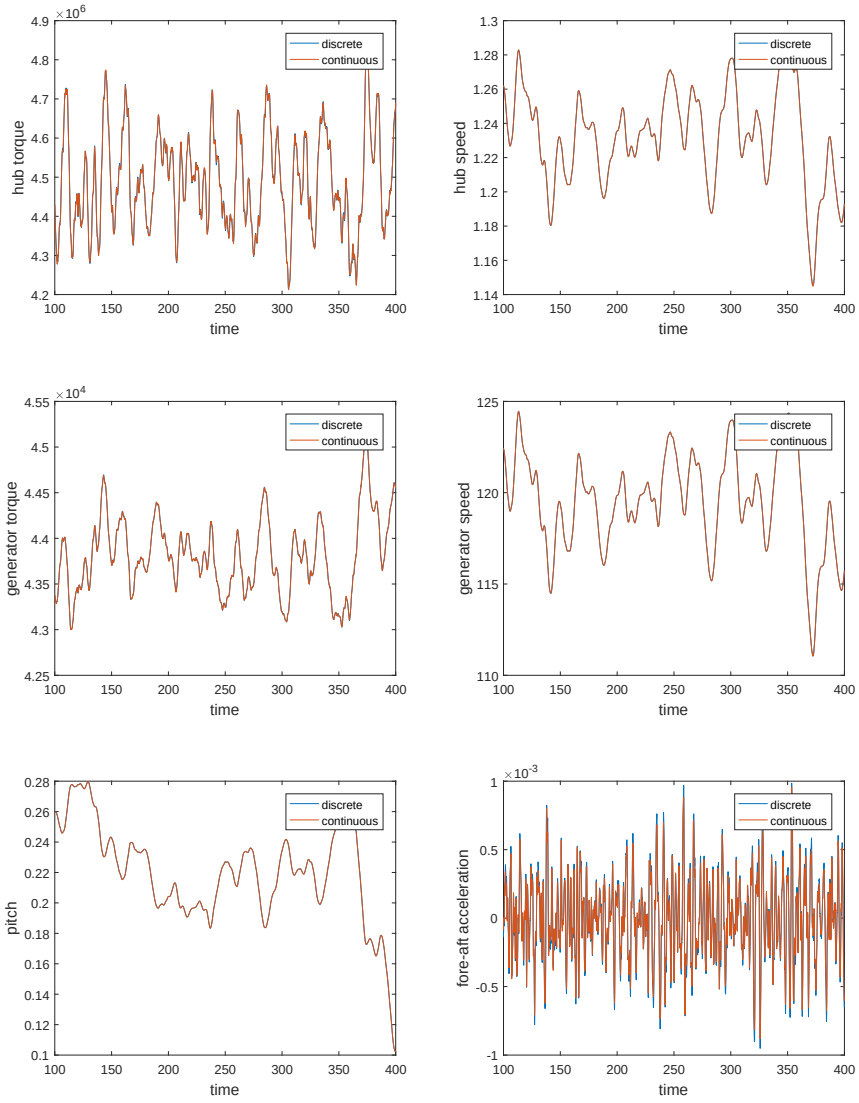


Figure 3: Continuous vs discrete measurements in time domain.

The results in Table 1 show that the discretised model is significantly faster. The improvement in speed becomes more predominant as the wind farm size

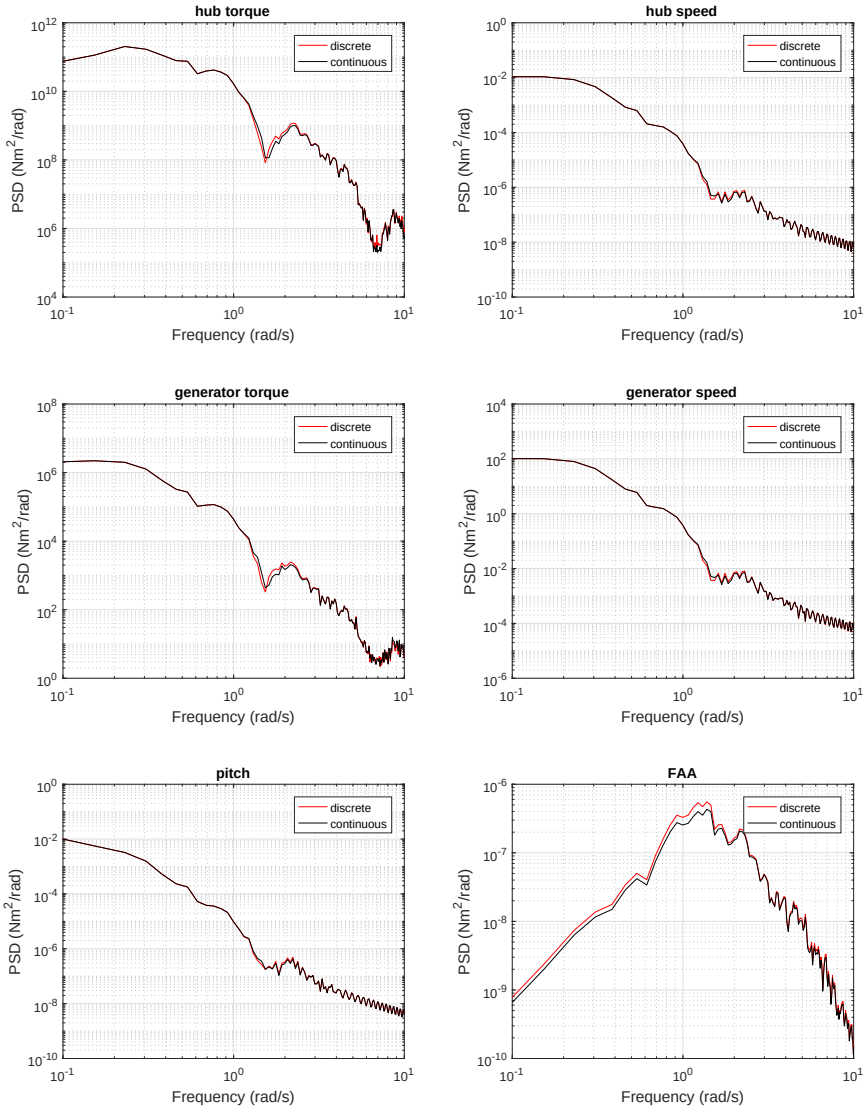


Figure 4: Continuous vs discrete measurements in frequency domain; FAA stands for fore-aft acceleration of the nacelle.

increases. A portion of model outputs from a discretised wind turbine in the farm model of 100 turbines that are especially important in the controller design

Table 1: Elapsed real time to simulate the continuous and discrete models for 500s

	1 turbine	10 turbines	50 turbines	100 turbines
Matlab/SIMULINK	31.22s	913.26s	too long	too long
C	0.75s	6.5s	35.2s	70s

– hub torque, hub speed, generator torque, generator speed, pitch and fore-aft acceleration of the nacelle (FAA) – are illustrated in comparison to those from the original continuous wind turbine model (reported in Section 2) in both the time and frequency domains in Figures 3 and 4, respectively. The blue and red plots in the figures are obtained from the discretised and continuous models, respectively. The time and frequency domain responses demonstrate that despite significant improvement in simulation speed, almost identical model outputs are produced; that is, the discretisation of the model and conversion to C results in significant improvement in simulation speed while keeping the model complexity intact.

5. Wind Farm Controller

A decentralised, scalable and flexible wind farm controller, capable of providing fast and accurate control of the wind farm power output to follow the wind farm power demand as determined by the grid side operation requirements for the wind farm, is introduced in [11]. A summary of the wind farm controller is presented as follows.

It can be used to maximise the aggregated wind farm power output and/or to follow a reference for the aggregated wind farm power output, taking into account fatigue loading on each wind turbine without compromising the turbines' own full envelope controllers through enclosing them in an additional feedback. The wind farm controller requires that each variable-speed pitch-regulated wind turbine be equipped with the Power Adjusting Controller (PAC) [10] that has been developed to provide fully flexible operation of an individual turbine. The wind farm controller uses the PAC to adjust the power generated by each turbine and causes the wind farm power output to meet the grid requirements, taking account of the status and the operating state of each turbine.

The wind farm controller also requires that each turbine be equipped with a full envelope controller, for which the existing Supergen Wind turbine controller is used. The full envelope controller causes the turbine to track its design operating curve as depicted on the torque/speed plane in Figure 5; that is, a constant generator speed (i.e. 70 rad/s) is maintained in the lowest wind speeds (mode 1); the C_{pmax} curve is tracked to maximise the aerodynamic efficiency in intermediate wind speeds (mode 2); another constant generator speed (i.e. 120 rad/s) is maintained in higher wind speeds (mode 3); and in above rated wind speed, the rated power of 5 MW is maintained by active pitching (mode 4).

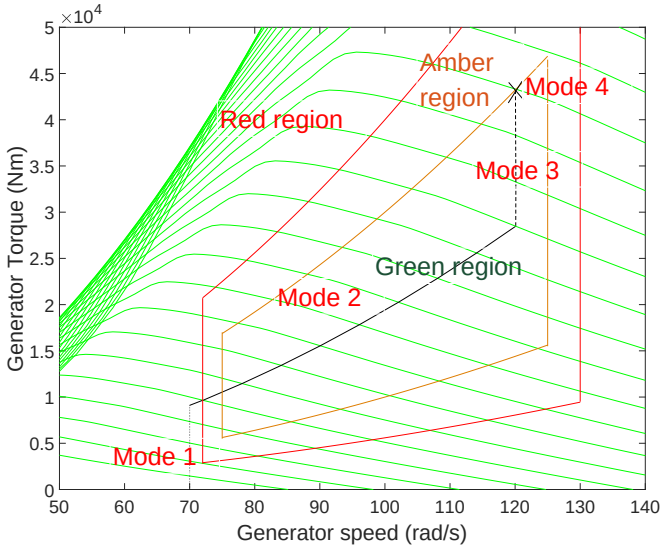


Figure 5: Design operating curve and operating regions (i.e. red, amber and green zones) defined by the thresholds over the full envelope operation on the torque/speed plane.

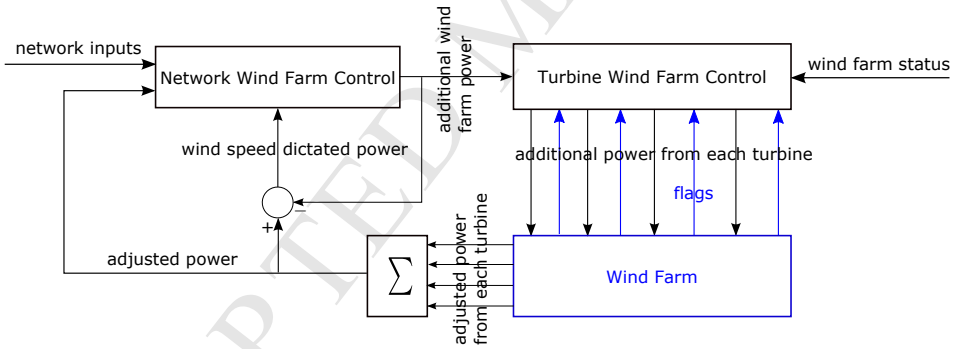


Figure 6: The structure of the wind farm controller.

As depicted in Figure 6, the wind farm controller has two elements, the Network Wind Farm Controller (NWFC) and the Turbine Wind Farm Controller (TWFC). The NWFC acts on information regarding the state of the power network to determine the required power output from the wind farm and hence the adjustment relative to the wind speed dictated wind farm power output, which would arise with no adjustment. The TWFC acts on information regarding the state of the wind farm and the turbines therein to allocate adjustments to each turbine relative to the wind speed dictated turbine power output.

The NWFC calculates ΔP (in Figure 6) using the following proportional-

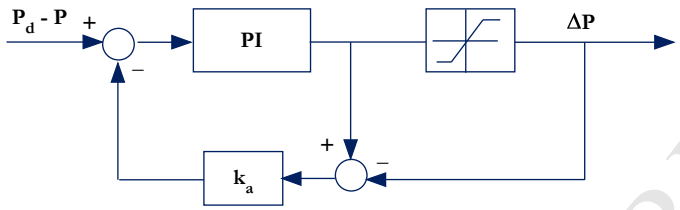


Figure 7: PI controller with anti-windup.

integral (PI) controller:

$$\Delta P(t) = k_p(P_d(t) - P(t)) + k_i \int (P_d(t) - P(t))dt \tag{29}$$

subject to the actuator constraints, where $P(t)$ and $P_d(t)$ denote adjusted power and demanded power, respectively, and k_p and k_i are the tuning parameters. The structure of the PI controller with anti-windup is depicted in Figure 7. In order to prevent integral windup, ΔP is limited to be less than 40 % of the rated power. Note that the PAC should not be utilised to curtail the power output by more than 30 % in real life [21]. k_a in the figure is the tuning parameter for the anti-windup loop.

Consequently, unadjusted power, P_m (the wind speed dictated wind farm power output that would arise with no adjustment) would be

$$P_m = P - \Delta P \tag{30}$$

The TWFC distributes ΔP to each turbine as ΔP_i based on flags, f_i (status of each turbine) and $\hat{f}_i$ (wind farm status as depicted in Figure 6) as follows

$$\Delta P_i = \frac{\Delta P \min(f_i, \hat{f}_i)}{\sum_{j=1}^{N_T} \min(f_j, \hat{f}_j)} \tag{31}$$

for $i = 1, \ldots, N$. The implication is that

$$\sum_{i=1}^{N_T} \Delta P_i = \Delta P \tag{32}$$

where N_T denotes the number of turbines in the wind farm. Equation (18) ensures that the power output from the wind farm tracks the demanded power even when the flags, hence ΔP_i, are being adjusted.

The simulation results [11] in Matlab/SIMULINK and BLADED demonstrate that the wind farm power output could be curtailed indefinitely to match the wind farm power demand while keeping each turbine in a safe operating

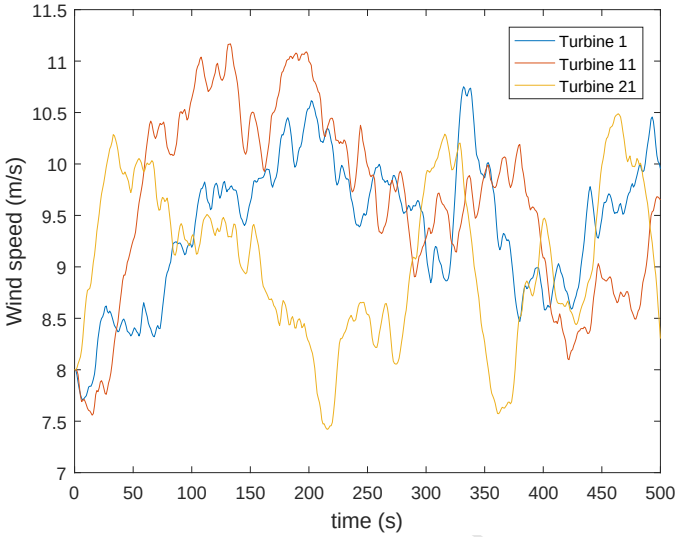


Figure 8: Wind speeds for three randomly selected wind turbines from the wind farm at a mean wind speed of 8 m/s.

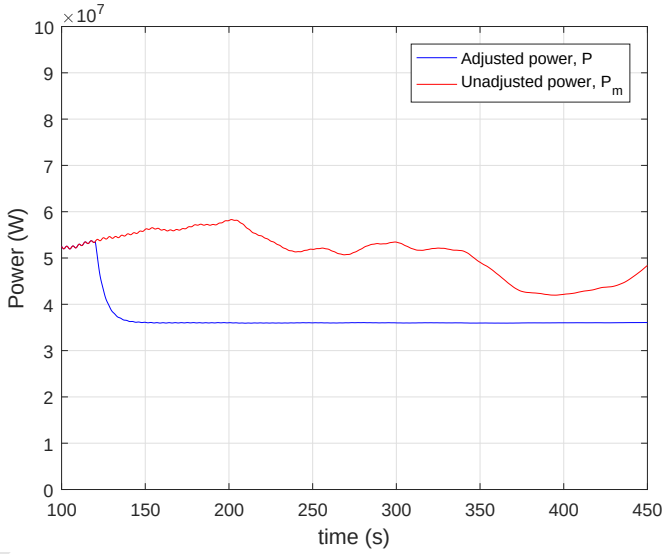


Figure 9: Adjusted power vs unadjusted power.

region by the use of thresholds defined on the torque/speed plane; that is, the thresholds divide the green, amber and red regions as illustrated in Figure 5. The allocation and reallocation of the power adjustments between the turbines

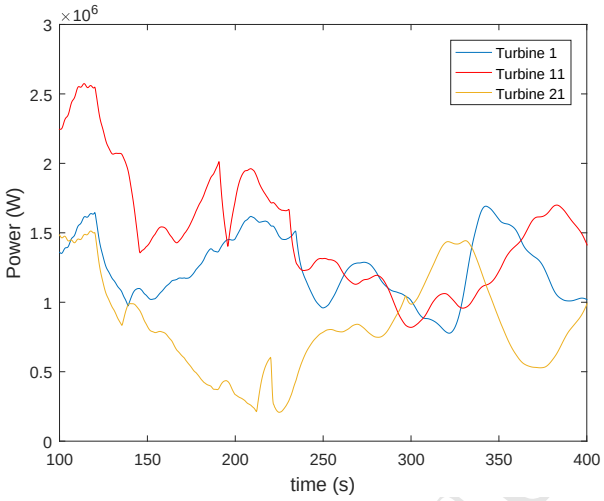


Figure 10: Power outputs from three randomly selected wind turbines from the wind farm.

occur smoothly avoiding the introduction of large transients, discontinuities and steps in the wind farm power output.

Since the wind turbine model used in the study [11] was continuous, the wind farm model could only consist of 10 turbines due to the computational effort. As a result of the discretisation and conversion to C reported in Section 3, the wind farm model of 30 turbines, with each turbine having almost the same complexity as before, is available. Importantly, this allows the wind farm controller to be tested against the wind farm model that is more realistic in terms of the wind farm size in the following simulation example.

In the simulation, the wind farm is required to produce a constant power of 36 MW at a mean wind speed of 8 m/s. The wind speed model introduced in [9] is used here which provides a suitable representation of the wind-field and wake propagation through the wind farm. The resulting wind speeds from turbines 1, 11 and 21 only, which are randomly selected, are depicted in Figure 8.

Adjusted power in blue is depicted against unadjusted power in red in Figure 9. The PAC is switched on at 120s past the transient period. Since it is the wind farm power output that is regulated, the individual power outputs from each turbine are still changing and being adjusted as depicted in Figure 10, in which the power outputs are presented from turbines 1, 11 and 21 only (randomly selected) while the aggregate wind power is almost constant as depicted in Figure 9. The turbines experiencing lower wind speeds, for much of the time, generate less than 1.2 MW (demanded power output divided by the number of turbines, N), while those experiencing higher wind speeds, for much of the time, generate more than 1.2 MW. Nonetheless, a constant wind farm power output of 36MW is produced as depicted in Figure 9.

The performance of each turbine can be observed from Figure 11, which

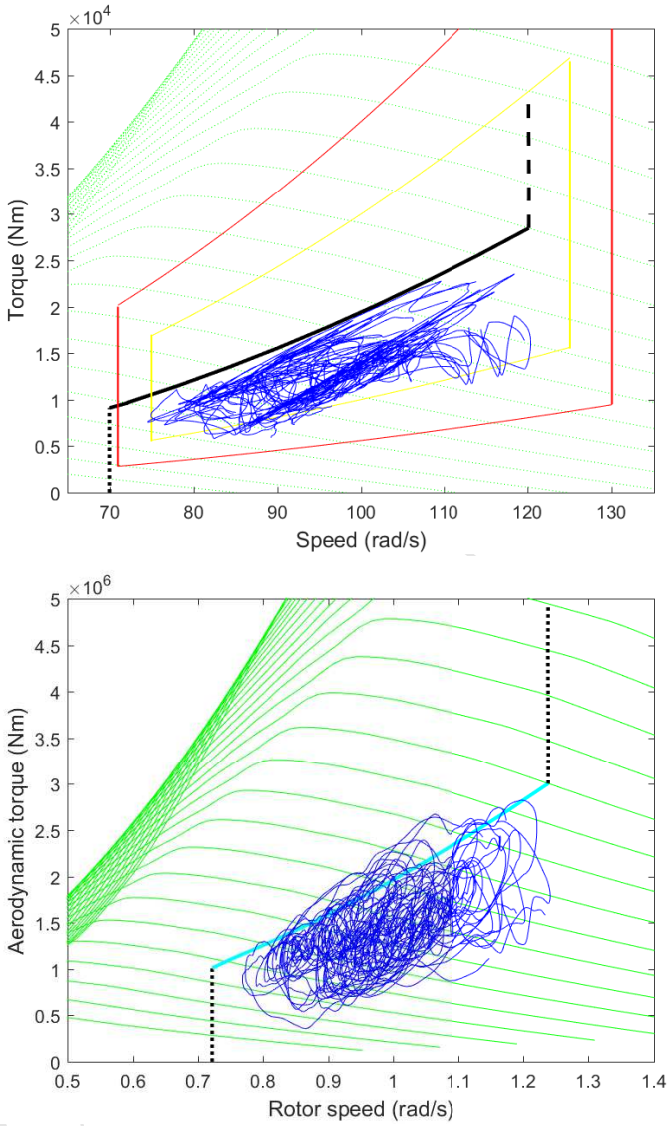


Figure 11: Behaviour of each turbine on the torque/speed plane.

depicts T_H (left sub-figure) and T_{Gen} (right subfigure) on the torque/speed planes from 100 to 450 s. The results demonstrate that the turbines operate within the green zone most of the time, allowing for the hysteresis loop. In more detail, when a turbine switches from the green zone to the amber zone, the turbine is allocated smaller adjustment in power, as the flag changes from 3 to 1, thereby causing the turbine to either slow down in moving towards the red zone or to return to the green zone. In this example, the turbines return to

the green zone, and the red zone is never reached.

Despite all the allocation and reallocation of P to each turbine as illustrated in Figure 11, in which the, a constant power of 36 MW is still maintained with little fluctuation as depicted in Figure 9.

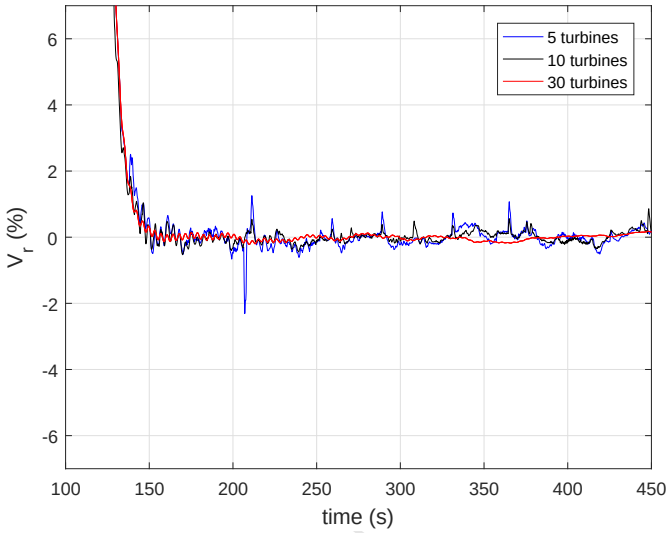


Figure 12: Fluctuations for 5, 10 and 30 turbine wind farms.

Due to turbulence, the steady state power output adjusted by the wind farm controller would always fluctuate. Fluctuations, $V_r(t)$, due to the difference between $P(t)$ and $P_d(t)$ (in %) is defined as [11]

$$V_r(t) = 100\frac{P(t) - P_d(t)}{P_d(t)} \tag{33}$$

where $P(t)$ and $P_d(t)$ denote adjusted power and demanded power, respectively.

$V_r(t)$ for the 30 turbine wind farm is depicted against $V_r(t)$ for a 10 turbine wind farm and a 5 turbine wind farm in Figure 12; 5 and 10 turbine wind farms are only exploited to provide a comparison, and its demanded power output is reduced by 1/6 and 1/3, respectively. The comparison demonstrates that $V_r(t)$ decreases as the wind farm size increases as expected, making the wind farm controller suitable for large wind farms.

As previously mentioned, although it is not feasible to apply the wind farm controller to a real-life wind farm at this stage, the original turbine model introduced in Section 2 has been fully validated by various researchers utilising high fidelity aeroelastic models developed in Bladed. This type of validation is a common practice not only academia but also in industry as DNV-GL (the company that develops Bladed) commissions and verifies real-life wind turbines; that is, a DNV GL Type Certificate provides the verification of real-life wind

turbine designs and provides independent proof of performance and safety in accordance with international standards and systems.

6. Conclusions

The Matlab model of the Supergen Wind 5 MW exemplar wind turbine, which has been used by various researchers over the last decade, especially within the Supergen Wind Consortium is improved in speed. This is achieved through prewarping, discretisation and conversion to C in order. Discretising optimally prevents the Simulink solver from non-optimally performing numerical integration for simulating the stiff drive-train module, and thus improves the simulation speed. Discretisation is also an essential prerequisite for conversion to C. Subsequently converting the discretised model to C, and then to CMEX for simulation in Matlab/SIMULINK, further improves the simulation speed. It allows the discretised model to be duplicated and extended to constitute a wind farm model without increasing the simulation time exponentially in contrast to the original continuous model. The simulation results demonstrate that even with one hundred turbine models included in a wind farm, the simulation speed remains at 70s. It is important again to emphasise that each turbine model is neither simplified nor compromised.

In the second part of the paper, the wind farm controller introduced in [11] is tested by application to a wind farm model of 30 turbine models, which is only possible because the turbine model has been improved in speed – in [11], the available wind farm model could only contain 10 turbines due to the computational effort and speed required for simulating each turbine model. The simulation results demonstrate that the wind farm controller performs satisfactorily when applied to the wind farm model of 30 turbines. The comparison of the simulation results to those from the situations in which the wind farm controller is applied to smaller wind farm models demonstrates that the fluctuation of the wind farm power output decreases as the wind farm size increases. The controller is therefore scalable and could be more valuable to larger wind farms.

Acknowledgment

This research was supported by Kyungpook National University Research Fund, 2017. This work was also supported by the EPSRC EP/N006224/1 “Maximising wind farm aerodynamic resource via advanced modelling (MAXFARM)” and FP-ENERGY-2013-IRP GAN-609795 “Integrated Programme on Wind Energy (IRPWind)”.

References

[1] E. Systems, Eltra, Wind Turbines Connected to Grids with Voltages above 100 kV: Technical regulation for the properties and the regulation of wind turbines, Tech. rep., Elkraft Systems and Eltra (2004).

[2] J. R. Kristoffersen, The Horns Rev Wind farm and the Operational. Experience with the Wind Farm Main Controller, in: Copenhagen Offshore Wind 2005, 26-28 October, 2005.

[3] D. Thompson, Mitigating size related limitations in wind turbine control, Ph.D. thesis, University of Strathclyde (2018).

[4] W. Leithead, B. Connor, Control of variable speed wind turbines: Dynamic models, International Journal of Control 73: 13 (2000) 1173 – 1188.

[5] P. M. O. Gebraad, M. J. Churchfield, P. A. Fleming, Incorporating atmospheric stability effects into the FLORIS engineering model of wakes in wind farms, Journal of Physics: Conference Series 753 (052004).

[6] P. Fleming, M. Churchfield, A. Scholbrock, A. Clifton, S. Schreck, K. Johnson, A. Wright, P. Gebraad, J. Annoni, B. Naughton, J. erg, T. Herges, J. White, T. Mikkelsen, M. Sjoholm, N. Angelou, Detailed field test of yaw-based wake steering, Journal of Physics: Conference Series 753 (052003).

[7] T. Lei, M. Barnes, S. Smith, S. Hur, A. Stock, W. E. Leithead, Using improved power electronics modeling and turbine control to improve wind turbine reliability, IEEE Transactions on Energy Conversion 30 (3) (2015) 1043–1051.

[8] A. Chatzopoulos, Full Envelope Wind Turbine Controller Design for Power Regulation and Tower Load Reduction, Ph.D. thesis, University of Strathclyde (2011).

[9] S. Poushpas, W. E. Leithead, Wind farm simulation modelling and control for primary frequency support, in: International Conference on Renewable Power Generation (RPG 2015), 2015.

[10] A. Stock, Augmented Control for Flexible Operation of Wind Turbines, Ph.D. thesis, University of Strathclyde (2015).

[11] S. Hur, W. E. Leithead, Adjustment of wind farm power output through flexible turbine operation using wind farm control, Wind Energy 19 (2016) 1667–1686.

[12] S. Winder, Analog and Digital Filter Design, Vol. 2, Elsevier, 2002.

[13] P. Pourazarm, L. Caracoglia, M. Lackner, Y. Modarres-Sadeghi, Stochastic analysis of flow-induced dynamic instabilities of wind turbine blades, Journal of Wind Engineering and Industrial Aerodynamics.

[14] W. E. Leithead, M. C. M. Rogers, Drive-train characteristics of constant speed hawt's: Part i - representation by simple dynamic models, Wind Engineering 20 (3) (1996) 149–174.

[15] L. Amos, University of strathclyde wind turbine model: Description, Tech. rep., Wind and Marine Energy CDT, University of Strathclyde (2017).

[16] W. E. Leithead, Effective wind speed models for simple wind turbine simulations, in: Proceedings of 14^{th} British Wind Energy Association (BWEA) Conference, Nottingham, 1992, pp. 321–326.

[17] R. Rafiee, M. Tahani, M. Moradi, Simulation of aeroelastic behavior in a composite wind turbine blade, Journal of Wind Engineering and Industrial Aerodynamics.

[18] R. Rafiee, M. Moradi, , M. Khanpour, The influence of material properties on the aeroelastic behavior of a composite wind turbine blade, Journal of Renewable and Sustainable Energy 8 (063305).

[19] W. Leithead, B. Connor, Control of variable speed wind turbines: Design task, International Journal of Control 73 (13) (2000) 1189 – 1212.

[20] M. L. Gala-Santos, Aerodynamic and wind field models for wind turbine control, Ph.D. thesis, University of Strathclyde (2016).

[21] A. Stock, Flexibility of operation, Supergen wind energy technologies consortium report, Department of Electronics and Electrical Engineering, University of Strathclyde (2013).

Highlights:

- The Matlab/Simulink model of a 5 MW exemplar wind turbine is improved in speed
- The model developed by Supergen Wind has been used for various projects in Europe
- The improvement is due to discretisation and conversion to C
- The discretisation employs both implicit and explicit methods with prewarping
- A wind farm controller is tested by application to the model