Control Oriented Modelling of a Wind Turbine and Farm

Sung-ho Hur[1] **and Bill Leithead**[2]

[1]School of Electronics Engineering, Kyungpook National University, South Korea
[2]Wind Energy & Control, University of Strathclyde, UK

E-mail: hur.s.h@ieee.org

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 improved, especially in speed, to facilitate wind farm modelling. Note that wind farm modelling usually involves duplicating wind turbine models, hence the speed of each turbine model is critical in wind farm modelling. The objective is achieved through various stages, including prewarping, implicit and explicit discretisation, and conversion to C. Simulation 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 not simplified or compromised.

1. Introduction

The Matlab/Simulink® (Matlab) model of the Supergen (Sustainable Power Generation and Supply) Wind 5 MW exemplar wind turbine was first introduced in [1]. It includes modules of aerodynamics, rotor dynamics, actuator dynamics, tower dynamics, drive-train, generator, etc, and has been updated/improved over the last decade and carefully validated against the high fidelity aero-elastic model, i.e. in DNV-GL Bladed (Bladed), of the same exemplar turbine. A through validation has been carried out a number of times and one of the most recent ones can be found in [2]. The Matlab model has been utilised for various different projects [3, 4], especially within the Supergen Wind Consortium, over the last decade.

Over the last several years, the model has also been utilised for developing wind farm models [5, 6, 7]. Wind farm models usually require the wind turbine model to be duplicated and, in turn, the simulation time could be increased exponentially – this is when each turbine model retains the full dynamics and is not simplified or compromised. Thus far, a wind farm model with more than 10 turbines has been considered impractical due to the simulation speed.

The objective of this work is to develop a wind farm model including as many as hundreds of the turbine models (without simplification), and to achieve this objective, we propose to optimally 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, and the non-optimal numerical integration can be prevented by optimally, manually discretising the model in advance, thereby avoiding the non-optimal numerical integration completely.

Stiff systems require a more stable "implicit" discretisation method, hence Heun's method is utilised. Heun's method is further modified to improve the result. Furthermore, stiff systems are known to potentially become unstable when discretised, which should be taken into account when discretising. This phenomenon is avoided by "prewarping" 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:

(i) optimal, manual discretisation to prevent the Simulink solver from non-optimally performing numerical integration for simulating the stiff drive-train module.

(ii) and conversion of the model to C, which requires the whole model, not only the drive-train module, to be discretised.

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.

In Section 2, the original continuous model of Supergen Wind 5 MW exemplar wind turbine is summarised, indicating various model parameters and variables that are included. The model is discretised and converted to C in Section 3. The simulation results are presented in Section 4, and conclusions are drawn in Section 5.

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

N	gearbox ratio	θ_R	angular displacement of rotor [rad]
R	rotor radius [m]		
T_H	hub torque [Nm]		
V	wind speed [m/s]		

2.1. Aerodynamics module

In-plane rotor torque is given by the following equation

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

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

$$\lambda = \frac{R\Omega}{V} \tag{2}$$

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

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

Thrust force is given by

$$F_t = \frac{1}{2}\rho\pi V^2 R^2 C_T(\lambda, \beta) \tag{4}$$

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

2.2. Rotor module

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

$$\begin{aligned}\ddot{\theta}_R = & -\omega_{es}^2[(\theta_R - \theta_H)cos\beta - (\phi_R - K_a\phi_T)sin\beta]cos\beta \\ & -\omega_{fs}^2[(\theta_R - \theta_H)sin\beta + (\phi_R - K_a\phi_T)cos\beta]sin\beta \\ & -(\omega_e^2 - \omega_{es}^2)(\Omega^2/\Omega_R^2)(\theta_R - \theta_H) + M_\theta/J\end{aligned} \tag{5}$$

$$\begin{aligned}\frac{JJ_T - J_C^2}{JJ_T + K_aJJ_C}\ddot{\phi}_R = & \omega_{es}^2[(\theta_R - \theta_H)cos\beta - (\phi_R - K_a\phi_T)sin\beta]sin\beta \\ & -\omega_{fs}^2[(\theta_R - \theta_H)sin\beta + (\phi_R - K_a\phi_T)cos\beta]cos\beta \\ & -(\omega_f^2 - \omega_{fs}^2)(\Omega^2/\Omega_R^2)(\theta_R - K_a\theta_T) + M_\phi/J \\ & + \frac{J_C/J}{J_T + K_aJ_C}(K_T\phi_T + B_T\dot{\phi}_T - hF_T)\end{aligned} \tag{6}$$

$$\begin{aligned}\frac{JJ_T - J_C^2}{JJ_T + K_aJ^2}\ddot{\phi}_T = & -\omega_{es}^2[(\theta_R - \theta_H)cos\beta - (\phi_R - K_a\phi_T)sin\beta]sin\beta \\ & +\omega_{fs}^2[(\theta_R - \theta_H)sin\beta + (\phi_R - K_a\phi_T)cos\beta]cos\beta \\ & +(\omega_f^2 - \omega_{fs}^2)(\Omega^2/\Omega_R^2)(\theta_R - K_a\theta_T) - M_\phi/J \\ & - \frac{1}{J_C + K_aJ}(K_T\phi_T + B_T\dot{\phi}_T - hF_T)\end{aligned} \tag{7}$$

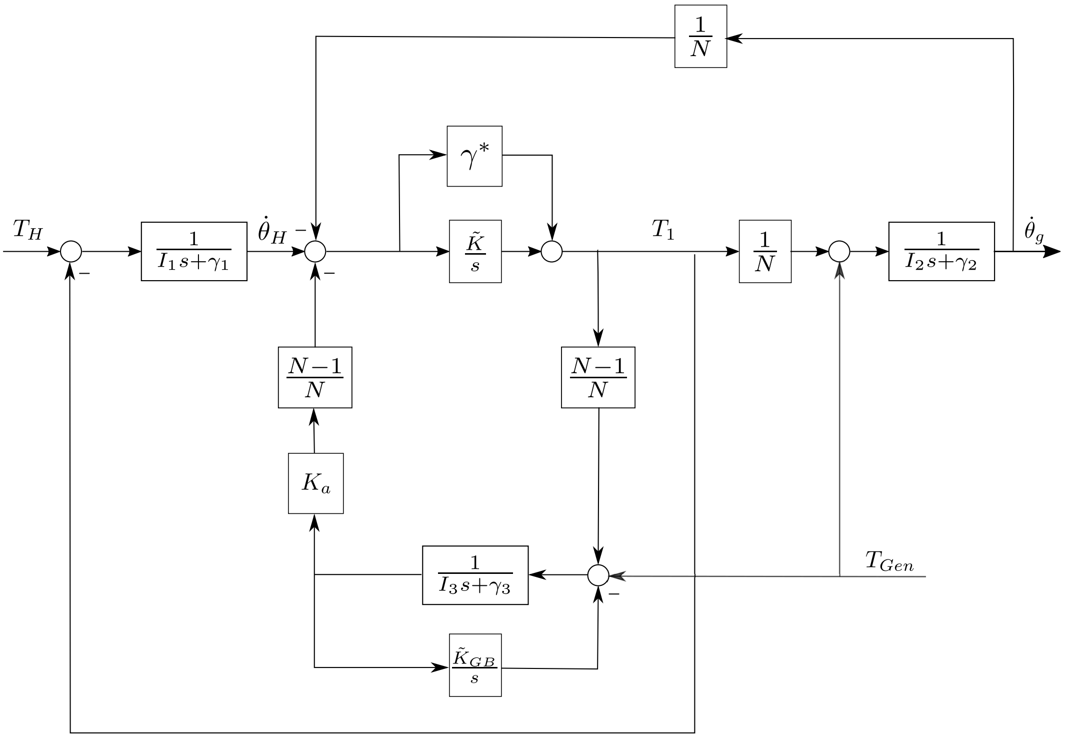

Figure 1: Drive-train module.

The hub torque dynamics is described by the following equation:

$$\begin{aligned}T_H =& J\omega_{es}^2[(\theta_R-\theta_H)cos\beta-(\phi_R-K_a\phi_T)sin\beta]cos\beta\\ &+J\omega_{fs}^2[(\theta_R-\theta_H)sin\beta+(\phi_R-K_a\phi_T)cos\beta]sin\beta\\ &-J(\omega_e^2-\omega_{es}^2)(\Omega^2/\Omega_R^2)(\theta_R-\theta_H)\end{aligned} \tag{8}$$

Combining (5), (6), (7) and (8), a 6th order nonlinear model can be obtained.

2.3. Drive-train and tower module

The drive-train module is depicted in figure 1. γ^* in figure 1 denotes drive-train material damping, a function of K_1, K_2, N, η, I_1, and I_2. The model depicted in the figure is a fifth order linear equation. The model is stiff causing the Simulink numerical solver to choose a non-optimal, large sampling step, thereby increasing the simulation speed. The model is optimally discretised in Section 3 to improve the speed. As previously mentioned, the model is further improved in speed by converting the discretised module, alongside the rest of the whole model discretised, to C.

2.4. Other dynamics and the turbine controller

The pitch mechanism is modelled using a second order transfer function.

Further dynamics and modules included in the model, but not presented in this paper include wind speed model [8], rotational sampling (i.e. 1P, 2P and 3P), and unsteady aerodynamics [6]. The commercially used Supergen Wind turbine controller for this turbine model is reported in [9] and has been discretised in [5]. The discretised controller is combined with the discretised turbine model for simulation throughout this paper.

2.5. Validation

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 [2].

3. Discretisation and Conversion to C

3.1. Discretisation

The continuous model reported in Section 2 includes various modules and dynamics, including aerodynamics module, rotor dynamics module and drive-train module. The drive-train module 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 optimal sampling step, resulting in a significant increase in simulation time. This is avoided here by optimally discretising the stiff drive-train module in advance and thereby not requiring numerical integration to be carried out at all during simulation.

The rest of the model, including the aerodynamics and rotors modules, are non-stiff and nonlinear, and the Forward Euler (FE) method is thus 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. Fortunately, however, because the modules are non-stiff, the use of "explicit" FE method instead of implicit methods is sufficient in any case.

Moving back on to the drive-train module, the poles that are close to the origin in the s-plane can 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 (i.e. to the left) 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 material damping of the drive-train module, γ^*.

Since the module is linear, the drive-train module reported in Section 2 can readily be represented in the following standard continuous state space form:

$$\begin{aligned}\mathbf{x}(t) &= A\mathbf{x}(t) + B\mathbf{u}(t)\\ \mathbf{y}(t) &= C\mathbf{x}(t) + D\mathbf{u}(t)\end{aligned} \tag{9}$$

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

Since one of the inputs is continuous and the other discrete, (9) is split into two equations, i.e. one with hub torque, $u_1(t)$, and the other with torque demand, $u_2(t)$. Subsequently, the two equations are discretised differently. By applying Heun's method to the first one as follows

$$\mathbf{x}(n+1) = \mathbf{x}(n) + \frac{T_s}{2}\left[(A\mathbf{x}(n+1) + Bu(n+1) + A\mathbf{x}(n) + Bu_1(n))\right] \tag{10}$$

the following discretised equation is obtained:

$$\mathbf{x}(n+1) = \underbrace{(I - \frac{T_s}{2}A)^{-1}(I + \frac{T_s}{2}A)}_{E}\,\mathbf{x}(n) + \underbrace{(I - \frac{T_s}{2}A)^{-1}\frac{T_s}{2}B}_{F}\,u_1(n+1) \tag{11}$$

$$\mathbf{y}(n) = C\mathbf{x}(n) \tag{12}$$

This equation is not in the standard state space form due to the implicit term $Fu_1(n+1)$. This term is, in fact, equivalent to the direct-feed term and hence can be moved to (12) such that

$$\mathbf{y}(n) = C\bar{\mathbf{x}}(n) + CFu_1(n) \tag{13}$$

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

$$\begin{aligned}\bar{\mathbf{x}}(n+1) &= E\bar{\mathbf{x}}(n) + (EF+F)u_1(n)\\ \mathbf{y}(n) &= C\bar{\mathbf{x}}(n) + (CF+D)u_1(n)\end{aligned} \tag{14}$$

Heun's method essentially allows the module to accept 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 step instead of the mean. Therefore, (11) is modified for the second input $u_2(n)$ as follows.

$$\begin{aligned}\mathbf{x}(n+1) &= E\mathbf{x}(n) + 2Fu_2(n)\\ \mathbf{y}(n) &= C\mathbf{x}(n) + Du_2(n)\end{aligned} \tag{15}$$

To recapitulate, when the input is aerodynamic torque, which is originally continuous, we use (14), and when the input is generator torque demand, which is discrete from the discrete controller, we use (15). 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.

$$\begin{aligned}\mathbf{x}(n+1) &= E\mathbf{x}(n) + G\mathbf{u}(n)\\ \mathbf{y}(n) &= C\mathbf{x}(n) + H\mathbf{u}(n)\end{aligned} \tag{16}$$

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 (9). G and H are given as follows:

$$G = \begin{bmatrix} EF_1 + F_1 & 2F_2 \end{bmatrix} \tag{17}$$

$$F = \begin{bmatrix} F_1 & F_2 \end{bmatrix} \tag{18}$$

$$H = \begin{bmatrix} J_1 & 0 \\ J_2 & 0 \end{bmatrix} \tag{19}$$

$$J = \begin{bmatrix} J_1 & J_2 \end{bmatrix}^T = CF_1 + D_1 \tag{20}$$

$$D = \begin{bmatrix} D_1 & D_2 \end{bmatrix} \tag{21}$$

(16) 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 FE 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 FE 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:

$$\mathbf{x}(n+1) = A\mathbf{x}(n) + B\mathbf{u}(n) \tag{22}$$

$$\mathbf{y}(n) = C\mathbf{x}(n) + D\mathbf{u}(n) \tag{23}$$

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 $\mathbf{x}(n) \in \mathbb{R}^{15}$.

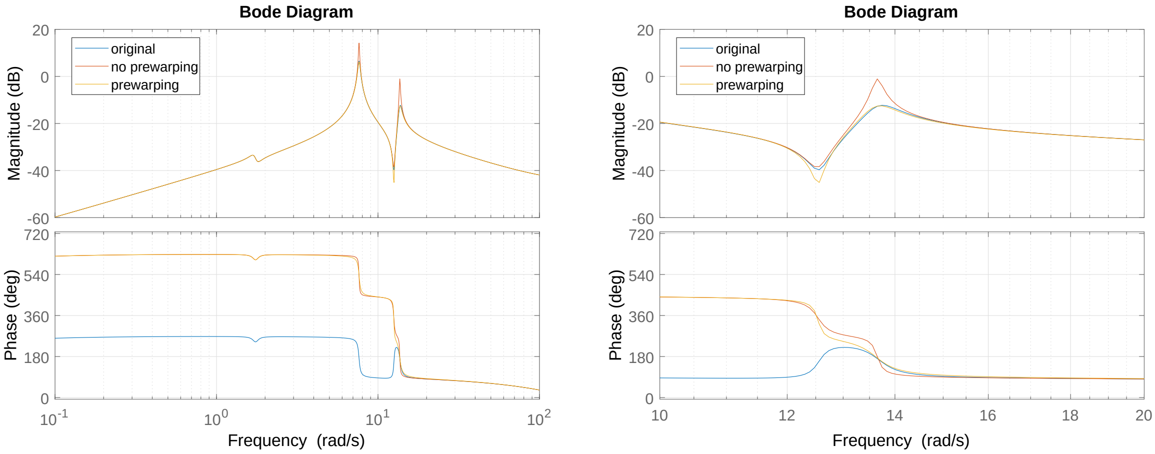

Figure 2: Discretisation and prewarping; the one on the RHS shows a zoomed version.

3.2. Prewarping

Shaft damping is altered prior to discretisation to ensure that the discretised module follows the characteristics of the original continuous module more closely and also to prevent the discretised model from becoming unstable due to the poles close to the y-axis in the s-plane ending up outside the unit circle in the z-plane. Figure 2 depicts the open-loop frequency responses of the wind turbine model including the full envelope controller. The blue plot is the response of the original continuous model and the red plot is the response of the discretised model – as a result of the discretisation described above – without prewarping. The yellow plot is the response of the discretised model that had also been prewarped before it was discretised. The responses in the figure illustrate that the discretised model exhibits more similar characteristics to the original continuous model if it is prewarped in advance. Further, without prewarping the increased peak at around 13.7 rad/s shown in the figure causes the discretised module to become unstable.

3.3. Conversion to C

The continuous model must be discretised to be converted to C, and this is one of the reasons that the model is discretised in Section 3.2 – another reason is to avoid non-optimal numerical integration during simulation as previously mentioned. Once the model has been discretised, it is ready to be converted to C. In more detail, the following procedure is conducted.

(i) The continuous model, including various modules, described in Section 1 is discretised, using the FE and Heun's methods.

(ii) The discretised model is written in an s-function; that is, the entire model is written in one s-function with all the parameters, inputs and outputs defined in Section 2.

(iii) The discrete s-function is rewritten in C.

(iv) 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 is equal to the original continuous model in terms of the model complexity and details; 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. Simulation Results

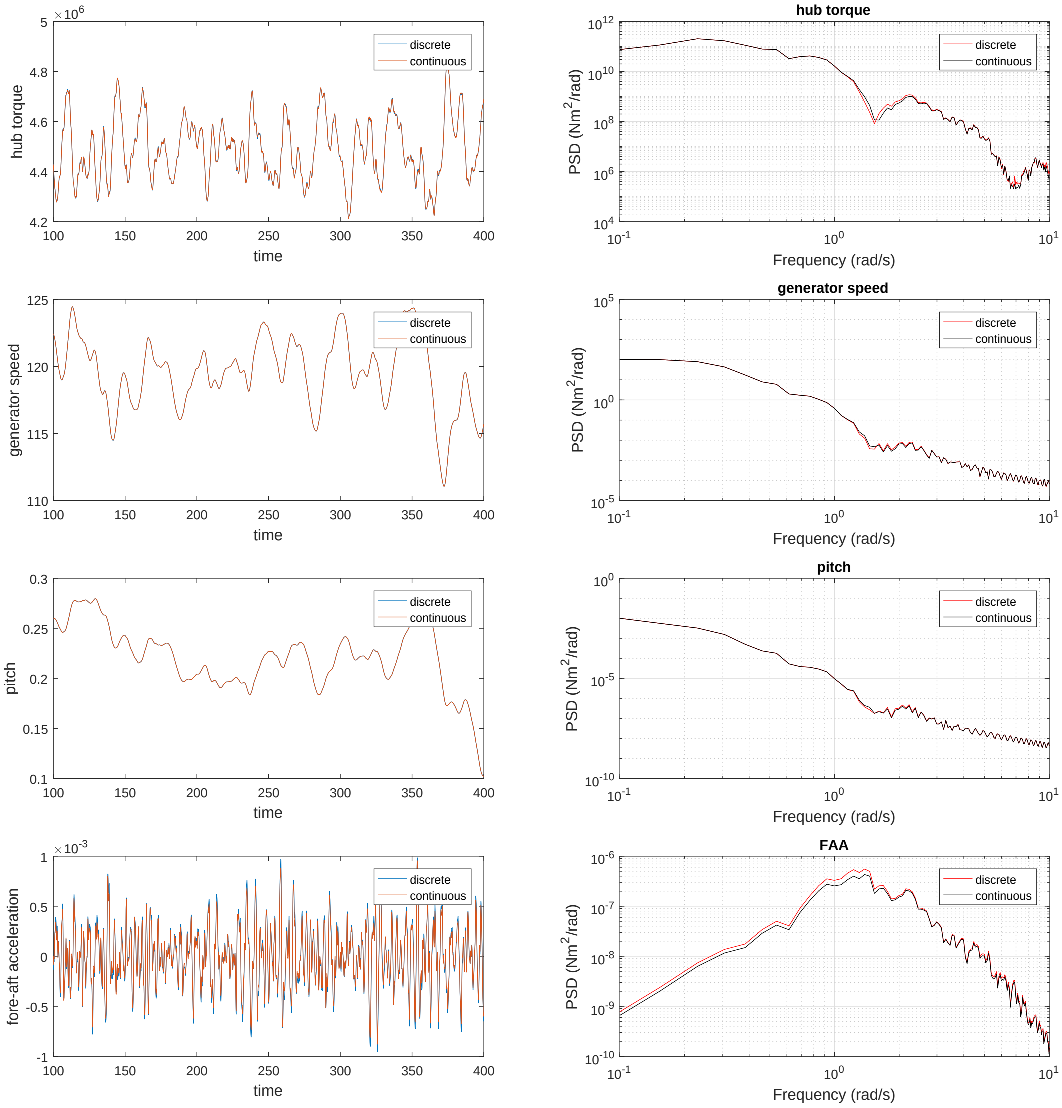

Figure 3: Continuous vs discrete measurements in time and frequency domain.

The discretised model from Section 3 is simulated in Matlab/SIMULINK in comparison to the original continuous model for 400 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 commercial full envelope controller that has been discretised in [5] and, and no wind farm controller is included at this stage of the project.

The results in table 1 show that the discretised model is significantly faster especially as the number of turbines included in the wind farm is increased; that is, the speed differential becomes greater as the wind farm size increases. A small portion of the model variables from one of the

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

discrete wind turbine models in the farm model are illustrated in comparison to the original continuous wind turbine model (reported in Section 2) in figure 3. 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, but the model complexity remains the same.

5. 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. The optimal discretisation based on Heun's method prevents the Simulink solver from non-optimally performing numerical integration for simulating the stiff drive-train module, thereby improving the simulation speed. The rest of the model is also discretised (using the FE method) since discretisation is also an essential prerequisite for conversion of the model to C. Subsequently converting the discretised model to C (and then to CMEX for simulation in Matab/SIMULINK) further improves the simulation speed.

As a result, the size of a wind farm model can be increased 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 is 70s. It is important again to emphasise that each turbine model is neither simplified nor compromised.

Acknowledgments

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] Leithead W and Connor B 2000 *International Journal of Control* **73: 13** 1173 – 1188
[2] Gala-Santos M L 2016 *Aerodynamic and wind field models for wind turbine control* Ph.D. thesis University of Strathclyde
[3] Lei T, Barnes M, Smith S, Hur S, Stock A and Leithead W E 2015 *IEEE Transactions on Energy Conversion* **30** 1043–1051
[4] Chatzopoulos A 2011 *Full Envelope Wind Turbine Controller Design for Power Regulation and Tower Load Reduction* Ph.D. thesis University of Strathclyde
[5] Poushpas S and Leithead W E 2015 *International Conference on Renewable Power Generation (RPG 2015)*
[6] Stock A 2015 *Augmented Control for Flexible Operation of Wind Turbines* Ph.D. thesis University of Strathclyde
[7] Hur S and Leithead W E 2016 *Wind Energy* **19** 1667–1686
[8] Leithead W E 1992 *Proceedings of* 14^{th} *British Wind Energy Association (BWEA) Conference, Nottingham*
[9] Leithead W and Connor B 2000 *International Journal of Control* **73** 1189 – 1212