EP/L014106/1

SUPERGEN Wind Hub

Deliverable:

D3.1 Report on Wind Farm Modelling

Delivered by:	University of Strathclyde		
Author(s):	Sung-ho Hur, Saman Poushpas, Lindsay Amos and Bill Leithead Wind Energy & Control Group		
Delivery date:	September 2019		
Distribution list:	Management Committee		

Prologue

In this report, the development of StrathFarm is described. StrathFarm is an analysis and design wind farm model and simulation tool with the following attributes.

- Wakes and wake interactions are modelled.
- Turbines are modelled in sufficient detail that tower, blade and drive-train loads are sufficiently accurate to estimate the impact of turbine and farm controllers on loads.
- Commercial standard turbine controllers are included.
- Wind farm controller and interface to turbine controllers are included.
- Very fast simulation of large wind farms are achieved, running in real time with 100 turbines on a standard PC.
- Choice of farm layout, turbines & controllers and wind conditions direction, mean wind speed and turbulence intensity is flexible.

StrathFarm is subject to ongoing development. During Supergen Wind Hub, the following tasks have been undertaken.

- The wind field models local to each turbine are improved. The approach is based on that reported in the PhD thesis, *Aerodynamics and Wind-Field Models for Wind Turbine Control*, M. L. Gala Santos.
- The turbine models are improved with a more detailed representation of the rotor and blades structural modes so that, together with the improved wind field models, all components of the blades, rotor and tower correctly are correlated correctly up to a frequency of 6P. The approach is based loosely on that reported in the MPhil thesis, *Individual Blade Control for Fatigue Load Reduction of Large-scale Wind Turbines*, V. M. Neilson.
- A choice of two wake models is now available, based on the Frandsen engineering model and WFSim, respectively. WFSim, developed at DTU, employs a 2D Navier-Stokes equation partially corrected for the neglected vertical dimension. Simulations, using WFSim are very slow but useful for validation of results.
- To achieve very fast simulation, all elements other than interfacing are now coded in C++ with the extensive use of dlls to facilitate different models for various elements. The continued use of MATLAB/Simulink with the very large number of dlls involved is now adversely affecting the speed of simulation and the portability of the software. To address these issues will require the MATLAB/Simulink interface to be dispensed with.

In addition to the above, a large number of other more minor improvements to StrathFarm have been made.

Following its remit to facilitate cooperation and collaboration, Supergen Wind Hub provided the cohesive element bringing together research from a number of programmes. The contributions from the following projects to the work reported here are gratefully acknowledged.

- EPSRC EP/L016680/1 DTC in Wind and Marine Energy Systems

- FP7-ENERGY-2013.10.1.6: 609795 IRPWind
- EPSRC EP/N006224/1 MAXFARM

In addition, background contributions from Adam Stock, Velissarios Kourkoulis, Lourdes Gala Santos and Victoria Nelson and from the following projects are gratefully acknowledged.

- EPSRC EP/G037728/1 DTC Wind Energy Systems
- EPSRC EP/H018662/1 Supergen Wind Energy Technologies Phase 2

CONTENTS

1. Wind Farm Modelling - Sung-ho Hur, Saman Poushpas and Bill Leithead
2. StrathFarm Wake Modelling Study: Part 1 - Lindsey Amos
3. StrathFarm Wake Modelling Study: Part 1a - Lindsey Amos
4. Control Orientated Modelling of a Wind Turbine and Farm – Sung-ho Hur and Bill Leithead
5. StrathTurb Models – Lindsey Amos

Supergen Wind

Title: Wind Farm Modelling

Sung-ho Hur, Saman Poushpas and Bill Leithead

Wind Energy & Control Group

Department of Electronic and Electrical Engineering

University of Strathclyde

June 2015

Overview

The main objective of this this document is to report on the wind farm model that is designed in Matlab/Simulink© to capture all the necessary dynamics required for the design of a wind farm controller and the load analysis of the wind turbines therein. It allows sufficiently fast simulation for iterative controller design task and contains a suitable wind-field model, providing a suitable representation of the wind-field and wake propagation throughout the wind farm.

Contents

Overview 2
1 Introduction 4
2 Wind Turbine Modelling 4
2.1 Wind Turbine Model 5
2.2 Full Envelope Controller 7
2.3 Power Adjusting Control 9
2.4 Implementation of the 5MW Supergen Controllers in C code and MATLAB C MEX S-Function 10
3 Wind Farm Modelling 11
3.1 Wind-field Model 13
3.1.1 Procedure of generating the turbulence time series 13
3.1.2 Wake modelling 14
3.2 Wind Turbine Model 22
3.2.1 High frequency turbulence data points 22
3.2.2 Generating Effective Wind Speed 25
3.3 Wind Farm Control Algorithm 25
3.4 Wind Farm Modelling Tool 25
4 Simulation of the Wind Farm 25
5 Summary 29
Bibliography 30

1 Introduction

A wind farm model [1] in Matlab/SIMULINK© that is suitable for fast simulation and for testing and designing a wind farm controller is introduced. It allows the investigation of the wind turbines' performance operating in wake conditions.

Most of the wind turbine and farm models available in the literature tend to possess dynamics that is too simplified and lacks the full envelope controller even though wind turbine control dynamics has significant impact on the operation of the wind turbines and farms.

Each wind turbine model introduced here includes not only the turbine, but also the full envelope controller, thereby allowing load analysis to be performed. The controller includes an additional feature, the Power Adjusting Controller (PAC) [2], which is used for adjusting the turbine's power output according to a power set-point change request from the wind farm control algorithm in a quick and safe operating manner. Essentially, the PAC serves as an interface between the wind farm controller and the wind turbines' full envelope controller to provide such a functionality in a safe and quick operating manner.

The wind farm model is capable of fast simulation even with a large number of wind turbines, and at the same time contains the necessary structural modes and dynamics of the wind turbines. Since the interaction between the rotor and the turbulent wind field induces loads on the wind turbine structure, a suitable representation of turbulent wind-field in the farm level and in the turbine level is included. Moreover, the wind-field model includes appropriate wake effects and their propagation throughout the wind farm making it suitable for wind farm controller design.

The wind farm control algorithm [1] is not reported in this document as the main purpose of this document is to report on the wind farm model. However, a wind farm control algorithm suitable for coordinating the wind turbines operation allowing flexible control of wind farm power output is included in the wind farm model to allow simulation to be performed.

A wind farm model construction tool is also developed to automatically build a wind farm model with a large number of turbines for users. The tool accepts user defined parameters and builds the corresponding wind farm model.

The wind turbine model including the full envelope controller and the PAC is presented in Section 2. The wind farm model including the wind field model, which includes the wake effect, is introduced in Section 3. The simulation results are reported in Section 4, and the summary of the work presented in this document is included in Section 5.

2 Wind Turbine Modelling

The Supergen 5MW exemplar wind turbine model (i.e. the nonlinear Simulink model that uses the parameters of the Supergen 5MW exemplar wind turbine) and its controllers including the PAC, which causes the turbine to adjust its generated power by a demanded amount relative to that dictated by the wind speed, constitute a wind farm model. In order to improve the simulation time, especially for wind farms with a large number of turbines, the PAC together with the full envelope controller is implemented in C++.

The resulting turbine model is illustrated in Figure 1.

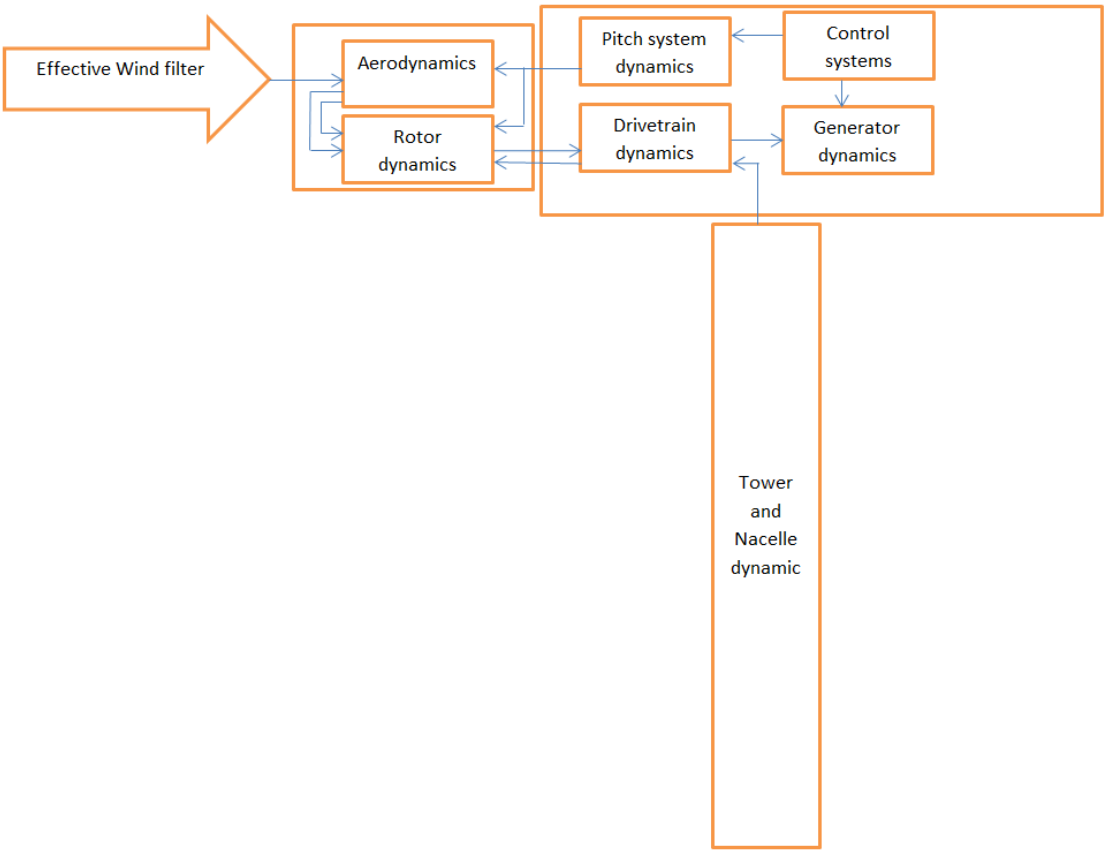


Figure 1: Wind turbine model [1].

2.1 Wind Turbine Model

The model includes the most significant structural dynamics relevant to controller design and is thus less complex in comparison to the high fidelity aero-elastic model of the same turbine that is available in DNV-GL Bladed (Bladed).

Two blade modes, two tower modes, actuator dynamics and a simplified drive-train model are included in each wind turbine model. These modes and dynamics are observed in Figures 2 and 3, in which the power spectra of fore-aft acceleration of the tower and generator speed are shown, respectively. Moreover, a nonlinear model of the rotor and wind-field dynamic interaction is included.

For further details of the turbine model, refer to [3].

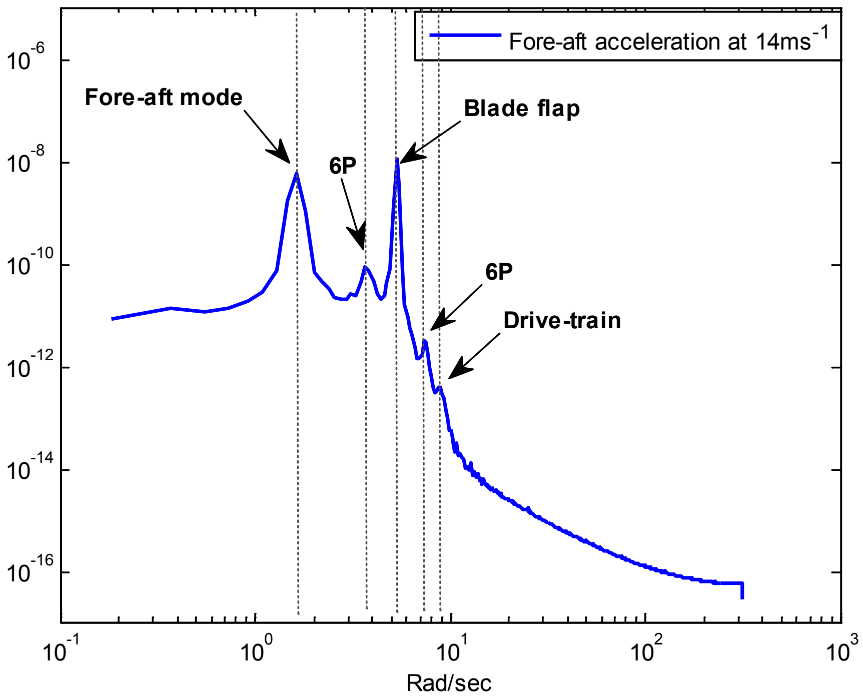


Figure 2: Power spectrum of the tower fore-aft acceleration [1].

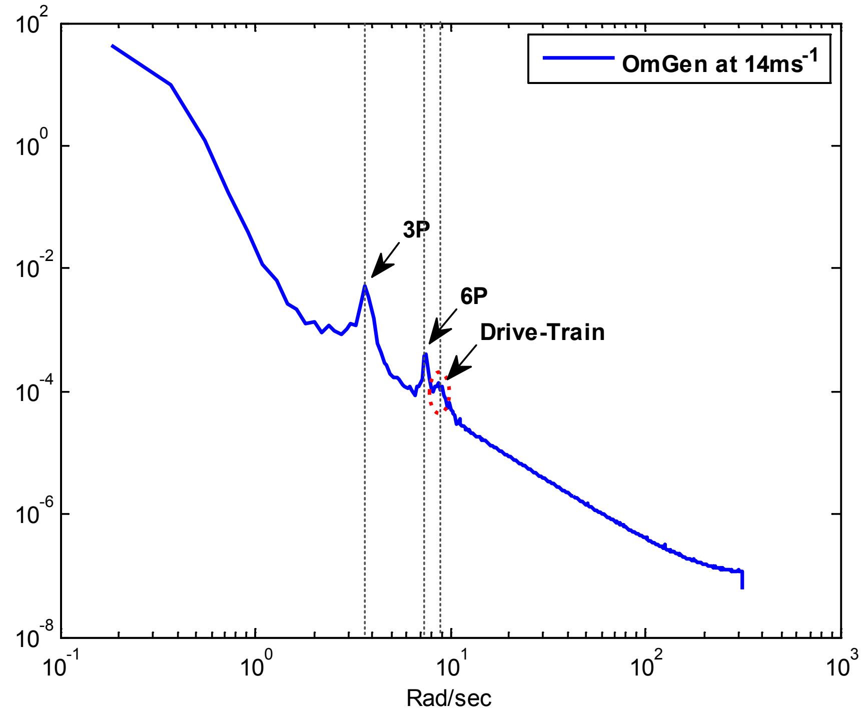


Figure 3: Power spectrum of the tower fore-aft acceleration [1].

2.2 Full Envelope Controller

The full envelope controller causes the turbine to track its design operating curve as depicted on the torque/speed plane depicted in Figure 4. More specifically, 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).

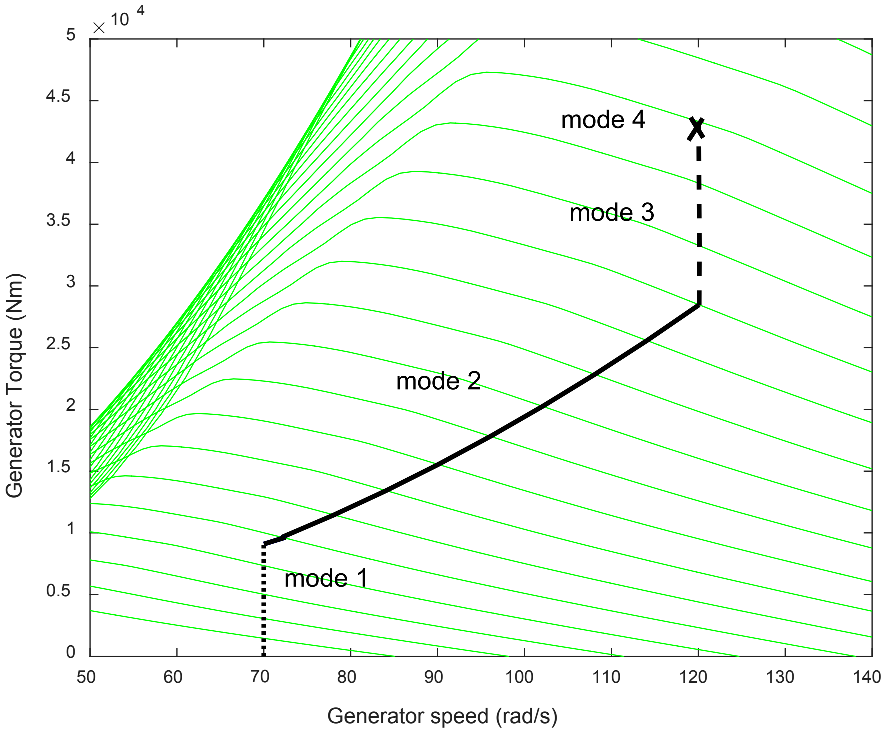


Figure 4: Design operating curve over the full envelope operation on the torque/speed plane.

The controller is based on a gain scheduled PI controller [3] using the Separability Theory [4]. Additionally, the drive-train damper and the tower damper are incorporated into the controller to damp the drive-train mode and the tower mode as depicted in Figures 5 and 6, respectively, in which the power spectra of generator speed and fore-aft tower acceleration with and without the dampers are depicted.

Further details of the full envelope controller can be found in [3].

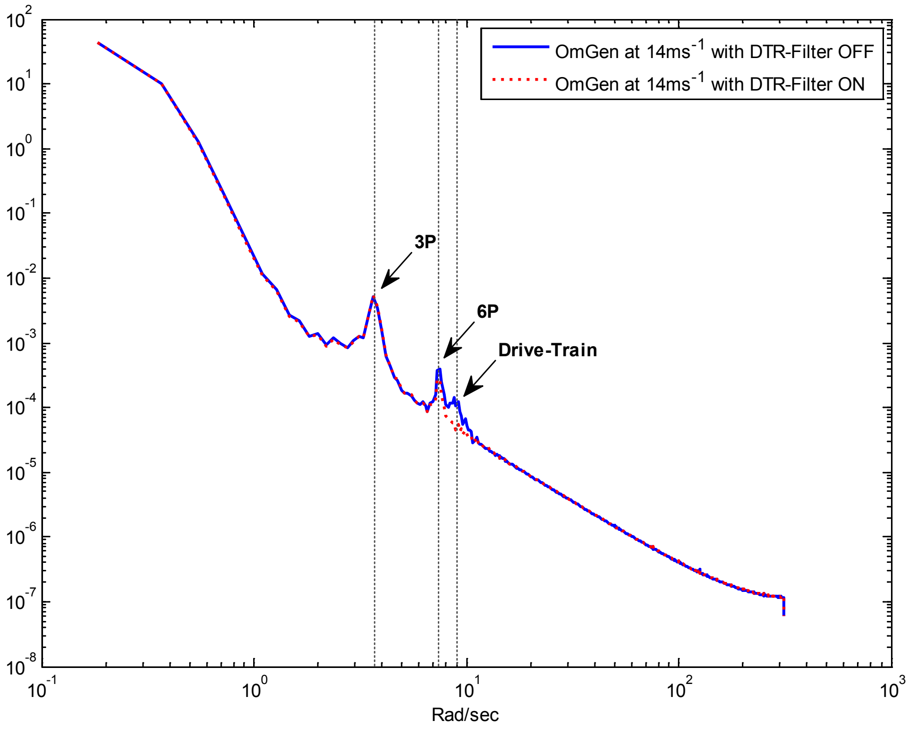


Figure 5: Power Spectral Density of generator speed [1].

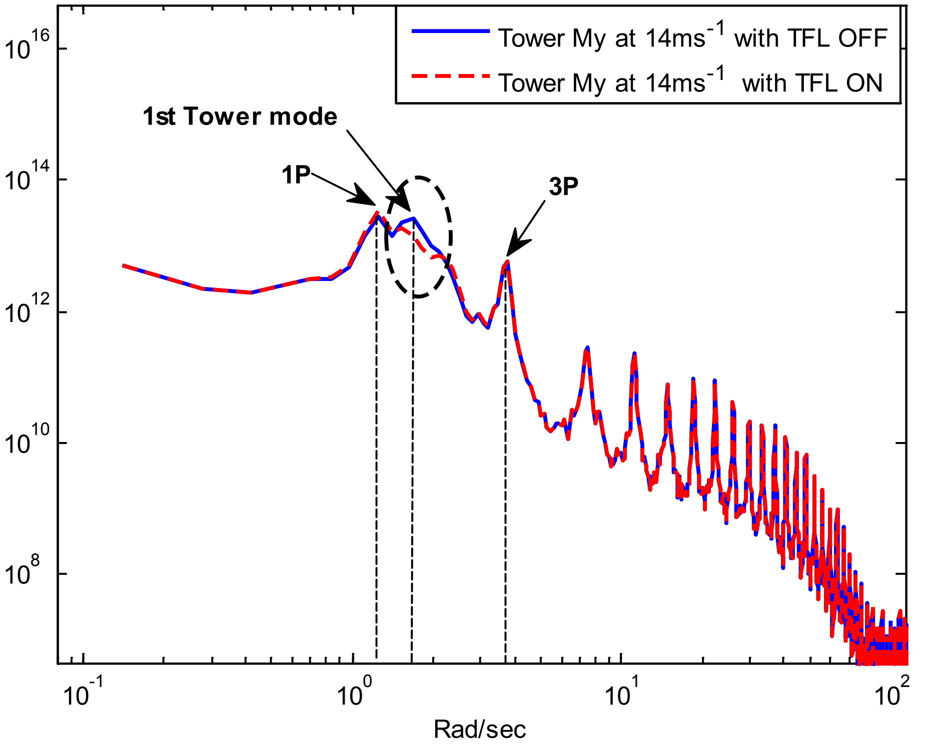


Figure 6: Power Spectral Density of fore-aft tower acceleration [1].

2.3 Power Adjusting Control

The PAC causes the turbine to adjust its generated power by a demanded amount relative to that dictated by the wind speed. As the PAC is essentially a feed-forward controller, jacketing the full envelope controller as depicted in Figure 7, it does not compromise the operation of the full envelope controller. Hence, redesigning or retuning of the existing full envelope controller is not necessary.

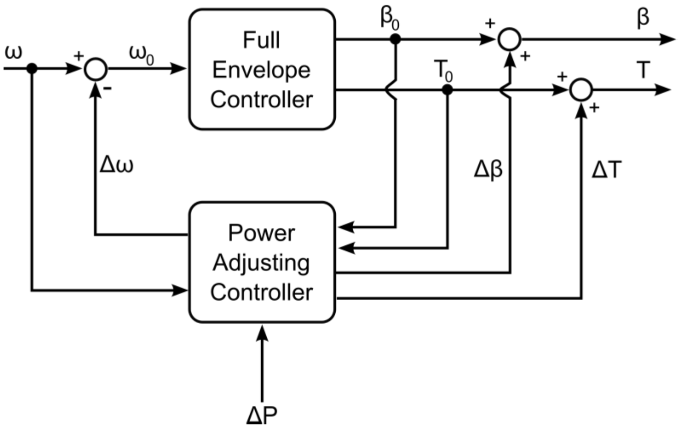


Figure 7: Structure of the PAC [1].

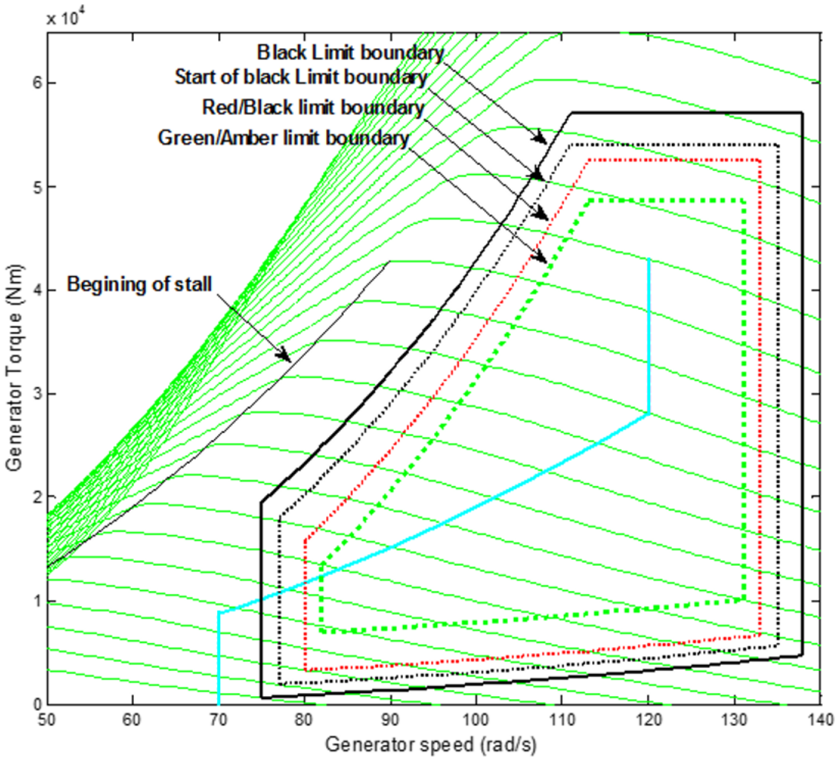


Figure 8: Operating boundaries for the PAC [1].

A set of PAC supervisory rules are defined to prevent the wind turbine from operating outside the safe operating boundaries when the PAC adjustments are being applied. The boundaries are illustrated in Figure 8.

2.4 Implementation of the 5MW Supergen Controllers in C code and MATLAB C MEX S-Function

The PAC together with the full envelope controller is implemented in C++ being fully equivalent to those installed on commercial wind turbines. The discretisation is performed using Tustin's bilinear transformation method. The resulting discretised C++ code is compiled into DLL using the C-MEX compiler included in Matlab, and the DLL is used by the s-function in Simulink to simulate the discrete controller.

The wind turbine model is a stiff system, including complex ordinary differentia equations (ODE). One of implicit variable step solvers thus needs to be used to simulate the model, but they are known to increase the simulation time significantly. To be able to use fixed-step explicit ODE solvers instead, such as the ode14x, to decrease the simulation time, the turbine model also needs to be discretised. However, the discretisation of the turbine model has not been performed to date and is left for future work.

The benefits of using the discrete controller can be summarised as:

- It improves the simulation time, thereby allowing the number of turbines in each wind farm model to increase.
- It allows decoupling of the sampling step of the simulation from that of the wind turbine controller, wind-field model and wind farm controller.
- Anti-windup loops can be implemented much more conveniently.
- The use of fixed-step solvers is allowed.

Figure 9 illustrates the process of compiling an external C MEX function written in C++ code and utilising the controller DLL in C MEX S-Functions in the Simulink model. "Controller.c" and "Controller.h" contain the discretised form of the wind turbine controller. "Controller1.c" is the wrapper function that is required by the compiler to generate the MEX file "Controller1.mexw64". The Simulink S-Function block calls the discretised wind turbine controller code at each iteration through "Controller1.mexw64".

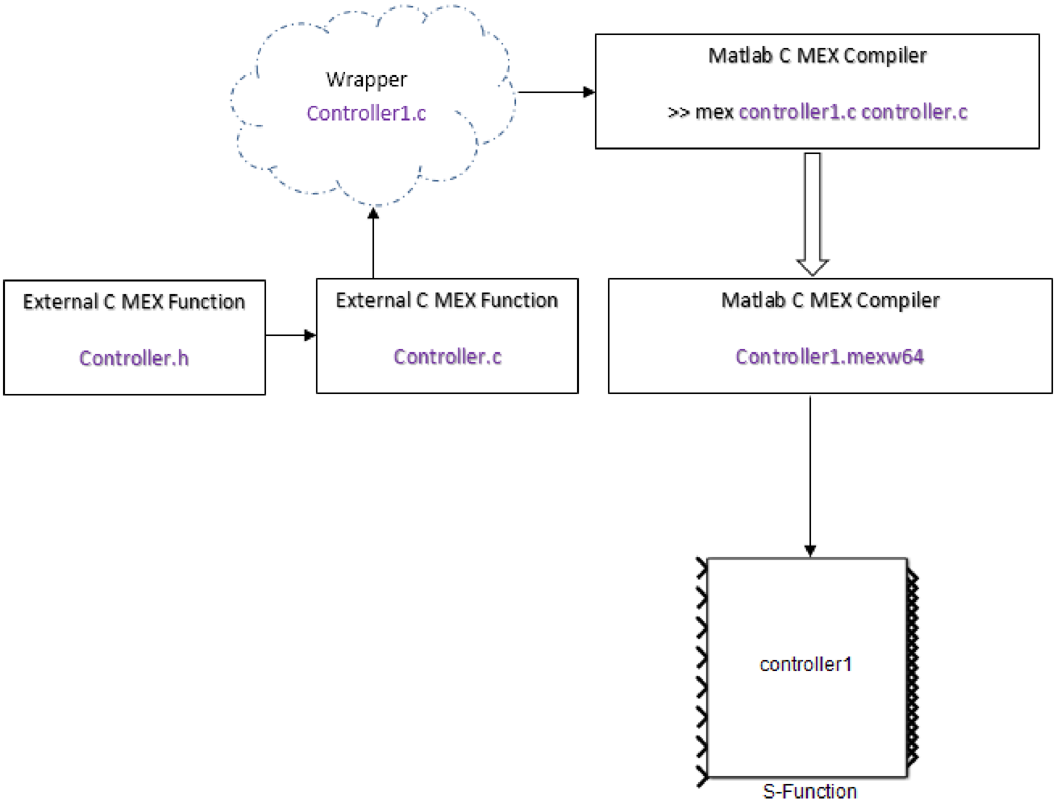


Figure 9: Compiling process of the controller written in C++ [1].

3 Wind Farm Modelling

A wind farm model that meets the following requirements is developed:

- It is fast even with a large number of wind turbines (e.g. more than 50 turbines) in order to facilitate wind farm controller design.
- Each turbine includes the full envelope controller and a PAC in addition to the turbine dynamics as discussed in Section 2.
- Wind field models representing turbulent wind field at the farm level and the turbines level are incorporated.
- Wake models that correctly represent wake propagation throughout the wind farm are incorporated.
- Wind farm control algorithm that coordinates the wind turbines' operation is included. This document is concerned with wind farm modelling, and wind farm control is beyond the scope of this document. However, it needs to be included to allow simulations to be performed.

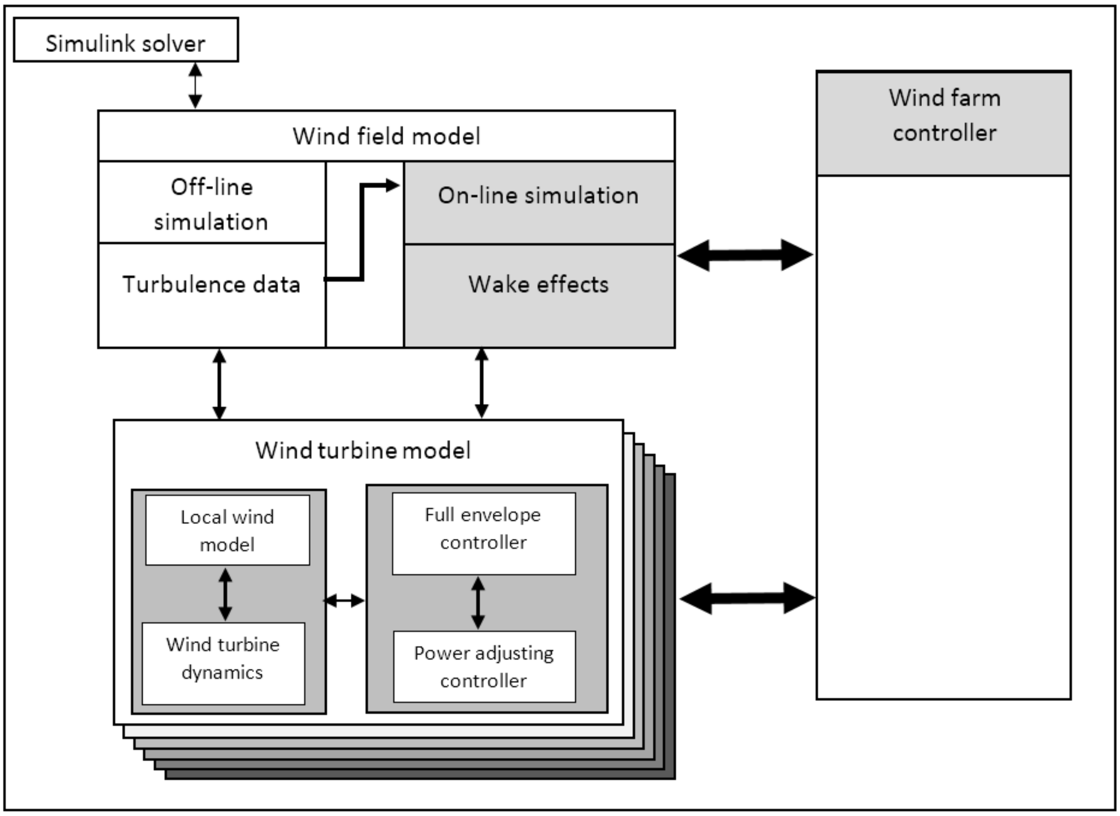


Figure 10: Schematic diagram of the wind farm model [1].

As depicted in Figure 10, the wind farm model consists of turbine models, each of which includes two main modules. The first module represents the wind turbine and the second module the full envelope controller including a PAC dedicated to each controller, as discussed in Section 3.

The wind-field model includes long length-scale aspects that correlate the performances of the turbines. These long length-scale aspects are the wind speed turbulence components, which are correlated at low frequency, and the wake effects and their propagation through the wind farm, which must also be considered at low frequency. The low frequency long length-scale aspects of the wind-field must be generated taking into account the correlation between the wind speeds at each turbine in the wind farm. However, the loads on the wind turbine also depend on higher frequency components in the wind-field. Therefore, the high frequency components of the wind-field is also added locally at each turbine. Similarly, to improve the simulation speed, the wind-field model is developed to minimise the computational effort.

Finally, in order to coordinate the operation of the wind turbines in the wind farm, a wind farm controller is required to communicate with the wind turbines through the interface controller, i.e. the PAC, at each turbine. The wind farm control is implemented in the discrete form to allow multiple control loops to operate at different sampling rates as required and also to improve the simulation time.

3.1 Wind-field Model

The correlated components of the wind field, including low frequency turbulence and wake propagation, have low frequencies, while the loads on a wind turbine tend to be at higher frequencies. This frequency separation is exploited here by including only the low frequency turbulence and wakes in the wind-field model while incorporating the high frequency turbulence into the wind turbine model, thereby improving the simulation time.

The wind turbines can be placed anywhere in the wind farm. The wind-field model for each turbine is represented by the longitudinal turbulence only, but to model propagation of the wakes through the wind farm, a more detailed representation of the wind-field model is included. More specifically, wake centres, wake diameters and wake deficit models are included in the wind farm model utilising a regular 2D grid, representing the lateral component of the wind-field as depicted in Figure 11. Density of the grid points defines the accuracy of the wake calculations but are kept to a minimum to avoid inversely affecting the simulation speed. The lateral turbulence data are also pre-calculated to be read during simulation. The wake meandering phenomenon is represented by calculating the wake centres at any distance behind the wind turbines producing the wake.

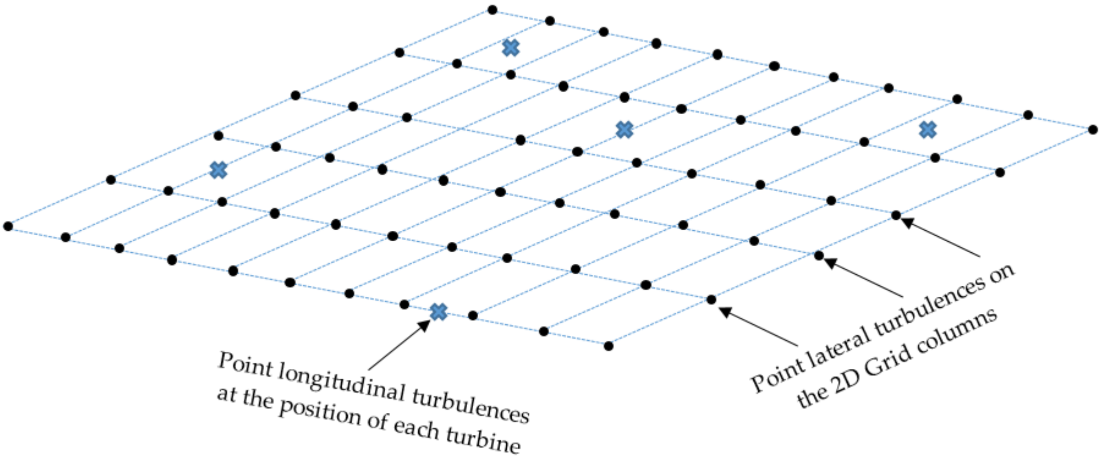

Point longitudinal turbulences at the position of each turbine
Point lateral turbulences on the 2D Grid columns

Figure 11: 2D grid layout [1].

3.1.1 Procedure of generating the turbulence time series

Veer algorithm generates N correlated turbulence time-series using Shinozuka cross spectral density function method based on single point PSD functions and for their coherence function for N wind turbines at their positions in the wind farm. Following the work presented in [5], the algorithm generates an N x N matrix of PSD functions including the coherence between the turbulent wind speeds at the N different coordinates within the wind-field. This $N \times N$ PSD function matrix contains both PSD functions on the diagonal elements and the cross spectral density functions on the off-diagonal elements. The off-diagonal elements are determined using a coherence function that describes the correlation between the turbulences at two point separated by a given distance.

The equations and further details are reported in [1].

3.1.2 Wake modelling

A suitable representation of the wake effects and its propagation through the wind farm are included in the wind farm model to facilitate controller design. A relevant approach for modelling the wake effects for fast wind farm model simulation is the kinematic methods. The most commonly used explicit wake modelling approach among them is the Frandsen analytical wake modelling method [6]. Using this method, wake effects are modelled taking into account wake centre, wake diameter and wake deficits, all as functions of distance and thrust coefficient.

Wind turbines could experience either a single wake or multiple wakes of neighbouring turbines depending on their position, and the Frandsen wake modelling method takes into account both. The single wake model considers the wake effect from only one turbine upstream, experienced by a turbine downstream. The multiple wake model combines the wake effects from all the turbines experienced by each turbine.

To compute and apply the wake effects at each turbine position, the wind turbines that induce wakes for each turbine need to be identified first. For instance, in a row of 5 wind turbines, with the one at the front facing the wind, the second turbine experiences wake effect only from the first turbine in the row ahead of it, while the fifth turbine may experience a combination of wake effects from the 4 turbines ahead. As the wake meanders through the wind-field, to identify the wind turbines affected by the wakes from the turbines upstream, the wake centre position and diameter of an upstream turbine at each turbine position must all be estimated. By mapping the wake centre and diameter of upstream turbines with the position and rotor diameter of the turbines downstream, the wind turbines in the wake can be identified.

3.1.2.1 Wake Centre

Wake centre is computed assuming an axisymmetric wake, meandering through the wind farm. Figure 12 shows the phenomenon of the wake meandering downwind of a wind turbine and the position of the wake centre as it evolves downstream. Figure 13 depicts a 2D layout of the wind farm, in which two wind turbines are positioned at (0,100) and (800,100) with 800m distance between them in a row aligned with the wind direction. The wind farm length and width are 330m and 300m, respectively.

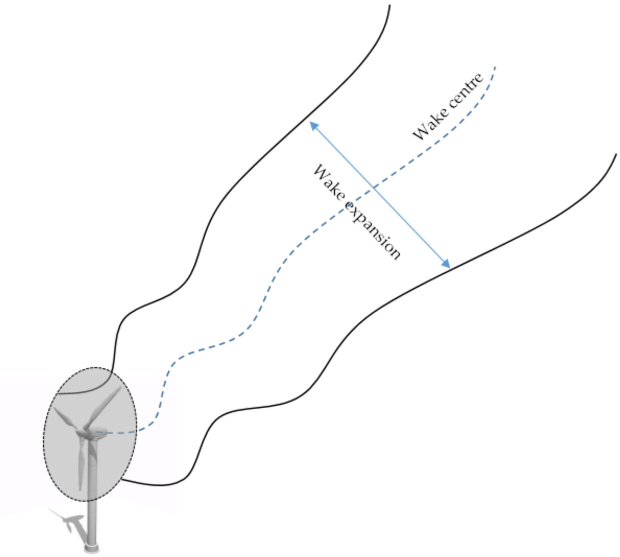


Figure 12: Wake meandering and wake centre [1].

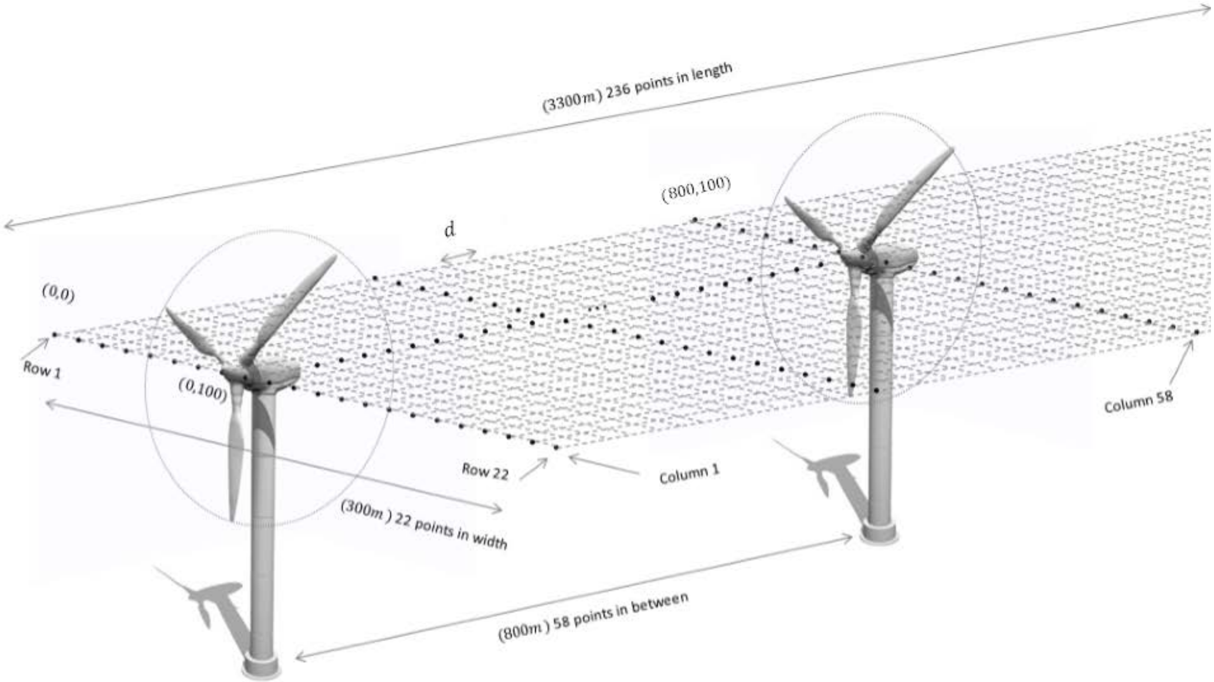

(3300m) 236 points in length
(800,100)
d
(0,0)
Row 1
(0,100)
Row 22
Column 1
Column 58
(300m) 22 points in width
(800m) 58 points in between

Figure 13: 2D grid layout of the wind farm [1].

The lateral grid lines represent the columns and the longitudinal grid lines represent the rows in the 2D grid mesh. For each grid point in the mesh a lateral turbulence time series is generated to be used for the calculation of wake centre and meandering. As mentioned earlier, the longitudinal turbulent wind speeds are only generated at the position of each turbine. Therefore, two longitudinal wind speed time series are saved to be used during simulation. The number of lateral turbulence time series depends on the number of grid points in the mesh. In order to compute the wake centre at any given distance downstream from a turbine, the longitudinal and lateral grid points between the upstream turbine and the given distance downstream are computed. The number of grid points in the longitudinal and lateral directions is calculated as

$$N_{long} = \frac{wind\, farm\, length}{d} = \frac{3300}{14} \cong 236 \tag{1}$$

$$N_{lat} = \frac{wind\, farm\, width}{d} = \frac{300}{14} \cong 22 \tag{2}$$

where d denotes the distance between the grid points. The lateral turbulence time series are computed for all the grid points in each column on the mesh.

The next step is to identify which lateral grid points are in the wake of a particular turbine. The wake radius in each column behind the turbine is computed and the grid points within the wake diameter are identified. Since the wake is assumed to be axisymmetric, the wake radius at a downstream distance x is computed as

$$R(x) = \sqrt{R_0^2 + \frac{R_0 x}{2}} \tag{3}$$

where R_0 denotes the rotor radius.

Figure 14 illustrates wake meandering radius at a distance downstream x from a turbine. The turbulence time series corresponding to the lateral grid points inside the wake at a distance x downwind of the rotor is utilised for computing the wake centre at that distance. The wake centre starts at the centre of the rotor just behind the turbine as shown in Figure 15. Therefore, in the particular situation, the lateral position of the turbine is the initial wake centre, i.e., $WC_0 = 100$, and the lateral displacement of the wake in each column between the turbine and distance x behind the turbine is computed as

$$WC = U_{L_{av}} * T_s + WC_0 \tag{4}$$

$$U_{L_{av}} = \frac{\sum_{i=1}^{n} u_{L_i}}{n} \tag{5}$$

$$WC_0 = WC \tag{6}$$

where $U_{L_{av}}$ is the average of the lateral velocity components inside the wake diameter in each column between the turbine and distance x behind the rotor, n the number of lateral grid points inside the wake diameter, and T_s the sampling time.

The lateral displacement in each column is computed and then added to the previous wake centre, which was computed in previous column to determine the wake centre at distance x behind the rotor. This is repeated for all columns between the wind turbine and distance x behind the turbine. The value of the wake centre at previous column updates the initial wake centre value WC_0 for the next iteration. Therefore, for the wake centre calculation in the next column, the previously computed wake centre becomes the starting point.

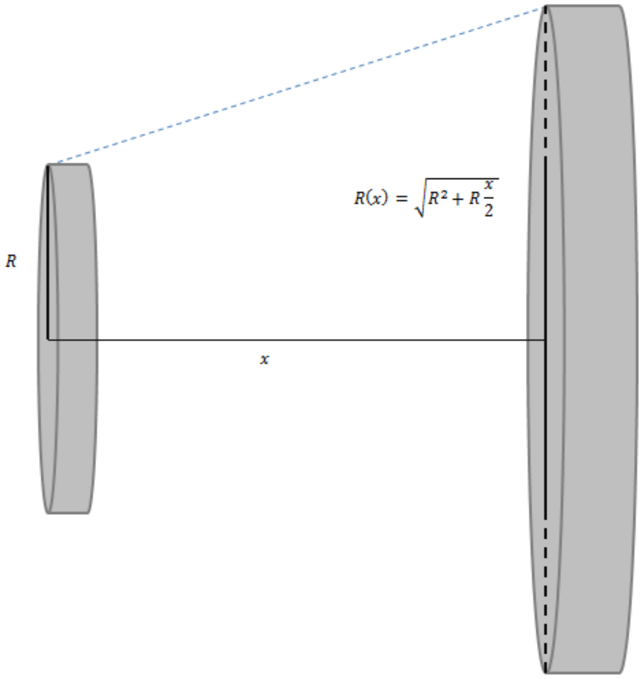


Figure 14: Wake meandering distance [1].

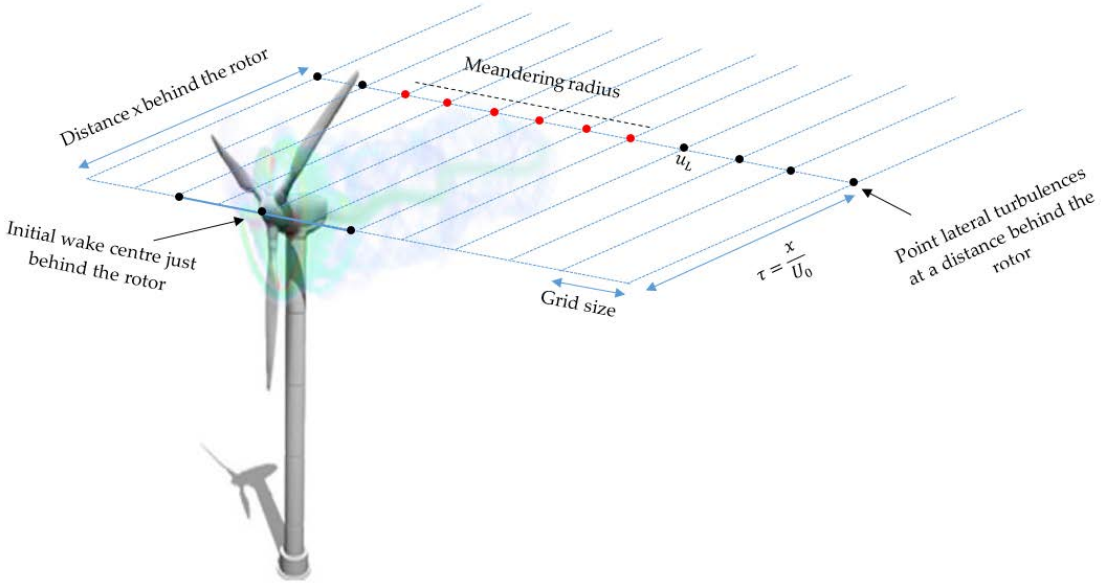

Distance x behind the rotor
Meandering radius
u_L
Initial wake centre just behind the rotor
Grid size
$\tau = \frac{x}{U_0}$
Point lateral turbulences at a distance behind the rotor

Figure 15: 2D grid layout with grid points inside the meandering radius at distance x [1].

For instance, if the number of columns in the longitudinal direction between two turbines is 10, the wake centre for 10 columns in between will be computed and added together to derive the wake centre at the lateral position of the downstream turbine.

3.1.2.2 Wake Diameter

The wake diameter is also computed based on the Frandsen method [6]. The wake is represented as a rectangular distribution of the flow speed. The thrust coefficient is related to the induction factor in the wake as follows

$$C_T = a(2-a), a = 1-\sqrt{1-C_T} \quad CT < 1 \tag{7}$$

The wake cross section area in the wake corresponding to this C_T coefficient is therefore

$$\frac{A_a}{A_0} = \frac{1-\frac{a}{2}}{1-a} \tag{8}$$

where A_0 denotes the swept area of the rotor and A_a the cross section area in the wake. It is assumed that the wake expands as soon as it passes the rotor, therefore for distance $x = 0$ behind the rotor the wake cross section is assumed to be A_a. By combining equations (7) and (8), we derive

$$A(x=0) = A_a = \beta A_0 \tag{9}$$

$$\beta(t) = \frac{1}{2}\frac{1+\sqrt{1-C_{T_i}(t-\tau)}}{\sqrt{1-C_{T_i}(t-\tau)}} \tag{10}$$

For increasing value of x and small wake flow speed, deficit wake flow speed is approximated as

$$\frac{U}{U_0} \approx 1 - \frac{1}{2}C_T\frac{A_0}{A} \tag{11}$$

The Frandsen method adopts an expression for the wake expansion as a function of distance such that equation (11) is still valid. At each sampling time, the wake expansion at a downstream distance is calculated based on the thrust coefficient of the upstream turbine, the distance downstream and the deficit delay between the turbines as follows

$$WD_j(x_{i,j}, t) = \left(\beta(t)^{\frac{k}{2}} + \alpha\frac{x}{D_0}\right)^{\frac{1}{k}} D_0 \tag{12}$$

where $WD_{i,j}(x_{i,j}, t)$ is the wake diameter at the lateral position of turbine j downstream, $k = 2$, $\alpha = 0.5$ and τ denotes the time it takes for the wake from a turbine upstream to reach to the lateral position of a turbine downstream.

3.1.2.3 Wake Deficit

Once the wake centre and diameter are computed, the wind turbines in the wake can be identified. By comparing the lateral position and rotor diameter of a wind turbine with the wake centre and diameter of the upstream turbines, the turbines inducing the wakes can be identified. The wake deficit for each turbine is computed either as a single wake or as a combination of multiple wakes from turbines upstream, according to the Frandsen single and multiple wake modelling methods. The wake deficits are computed based on the wake centre, wake diameter, thrust coefficient and the wake deficit delay at the position of each wind turbine downstream.

Using equation (12) for the wake expansion area as a function of distance, the wake deficit stemmed from a single turbine is defined as

$$Def_j(t) = 1 - \frac{1}{2}\frac{CT_i(t-\tau)D_0^2}{WD_j(t)^2} \tag{13}$$

where $Def_j(t)$ is the velocity deficit at turbine j due to the wake from turbine i, and $WD_j(t)$ denotes the wake expansion diameter of turbine i seen by turbine j at time t, which is computed using equation (12). As it takes some time for the wake to reach the downstream turbine, the deficit delay is also taken into account in the calculation. Therefore, deficit is computed based on the thrust coefficient of the upstream turbine at time $t-\tau$, where τ denotes deficit delay between the turbines. To incorporate deficit delay, thrust coefficient is delayed appropriately to represent the deficit delay between the turbines. Hence, a suitable propagation of the wake through the wind farm is modelled.

The calculation of single wake can be expanded to calculate multiple wakes. In a wind farm with a large number of turbines, a combination of the wakes from upstream turbines affect the wind turbines in the wake as illustrated in Figure 16. Therefore, it is recommended the effects of multiple single wakes be combined at each turbine.

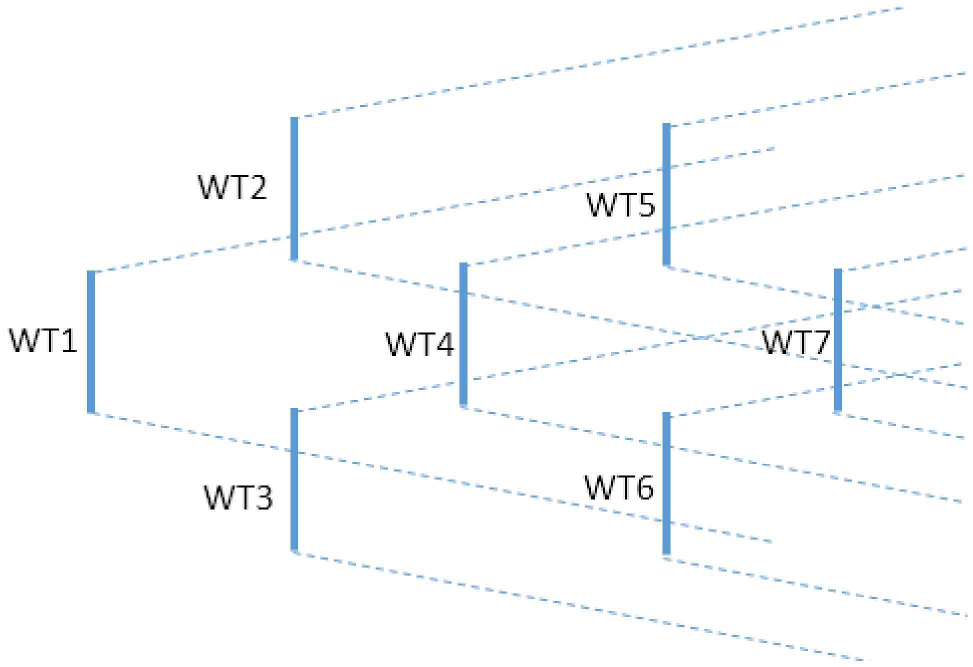


Figure 16: Multiple wake effects [1].

Various methods are available to combine multiple single wake effects at each turbine. The main methods, including sum of squares of velocity deficits, energy balance, geometric superposition, linear superposition, can be found in [95].

The multiple wake deficit combination based on the Frandsen method is carried out considering wind turbines in a single row parallel to the wind direction. This method is only used for the special cases where the wake deficit is computed as a combination of wake effects from upwind turbines in a single row. The multiple wake deficit is computed considering the momentum conservation in the control volume as follows

$$\rho A_{n+1}\, U_{n+1}(U_0 - U_{n+1}) = \rho A_n\, U_n(U_0 - U_n) + T \tag{14}$$

$$T = \frac{1}{2}\rho A_0 C_T U_0^2 \tag{15}$$

Substituting the thrust force expression into equation (14) yields

$$\frac{U_{n+1}}{U_0} = 1 - \left[\frac{A_n}{A_{n+1}}\left(1 - \frac{U_n}{U_0}\right) + \frac{1}{2}\frac{A_0 C_T}{A_{n+1}}\left(1 - \frac{U_n}{U_0}\right)\right] \tag{16}$$

Considering wake deficit transport delays, the multiple wake deficit at turbine $n + 1$ is computed as a combination of single wake deficit from the closest neighbouring turbine n and the wake deficit effects of all other turbines, on wind turbine $n + 1$.

$$Def_{n+1} = 1 - \left[\underbrace{\frac{WD_n(t-\tau)^2}{WD_{n+1}(t-\tau)^2}\left(1 - Def_n(t-\tau)\right)}_{effect\ of\ other\ turbines} + \underbrace{\frac{1}{2}\frac{C_{T_n}(t-\tau)D_0^2}{WD_{n+1}(t-\tau)^2}\left(1 - Def_n(t-\tau)\right)}_{effect\ of\ closest\ turbine}\right] \tag{17}$$

As depicted in Figure 17, the wake effects at turbine WT_{n+1} stems from that from WT_n and the other turbines in the row. The equation considers the wake deficit transport delays in the calculations. The wake deficit transport delay is the time taken by the wake to propagate from turbine WT_n to WT_{n+1}. This special case of combining the wake deficits of the wind turbines in a single row is useful for designing wind farm control algorithms with the purpose of array efficiency optimisation. However, since wind turbines in a wind farm may not be directly positioned in line, another method for combining multiple wakes should also be included in the wind farm model.

Since the wake deficit algorithm searches for the wind turbines producing wakes at the position of each turbine, regardless of their position in the wind farm, the wake combination can be obtained considering single wake effects at each turbine position. Therefore, at each turbine the wake effects from all turbines producing the wake are combined to obtain the resultant wake deficit. However, wake effects at a downwind turbine can result from a combination of partial wake from the upwind turbines as shown in Figure 18. Therefore, at each turbine the overlapping wake shadow area of the upwind turbines is computed and only a proportion of the wake deficit from the upstream turbine is considered in the wake combination calculations.

The wake overlapping area is computed using the area overlapped by two overlapping circles with known diameter and distance between them. The shadowing area is computed by

$$A_{shadow} = r^2\cos^{-1}\left(\frac{d}{2r}\right) - \frac{d}{4}\sqrt{4r^2 - d^2} + R^2\cos^{-1}\left(\frac{d}{2R}\right) - \frac{d}{4}\sqrt{4R^2 - d^2} \tag{18}$$

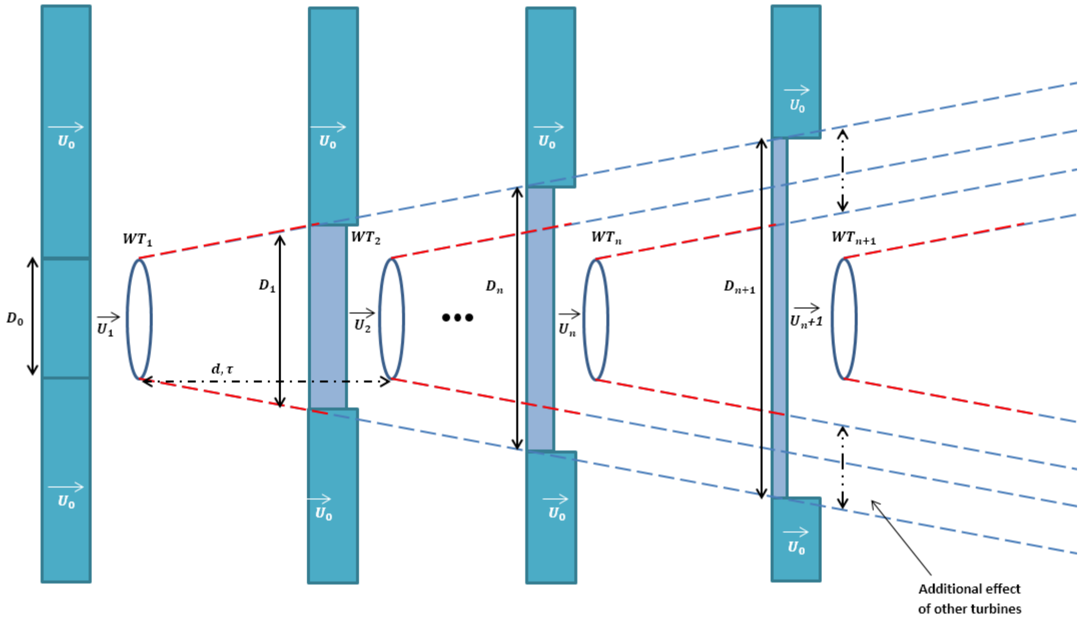


Figure 17: Effect of wake deficit [1].

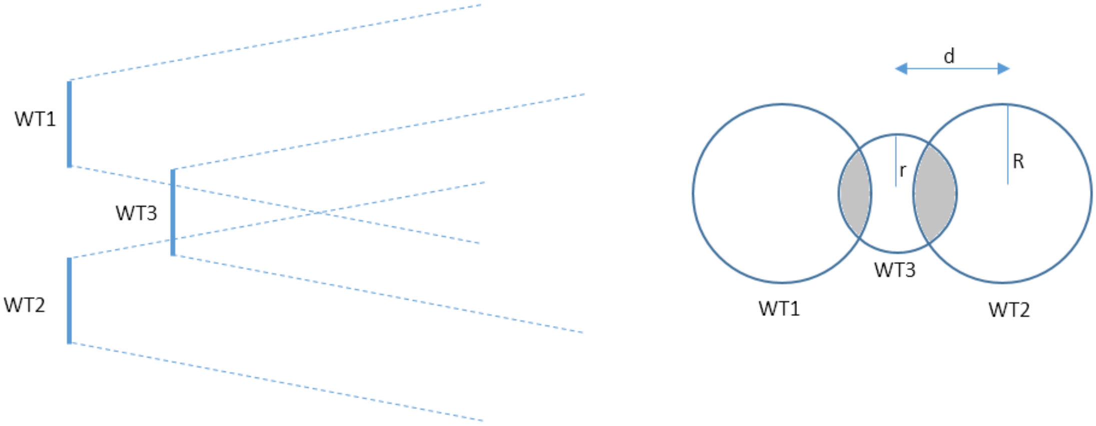


Figure 18: Partial wake shadowing effect [1].

Once the shadow area is computed, the effect of single wakes from the upstream turbines are combined according to the sum of squares of velocity deficit method.

$$def_j = \sqrt{\sum_{\substack{j=1 \\ j\neq i}}^{n} def_{i,j}^2 \frac{A_{shadowi,j}}{A_0}} \tag{19}$$

where, $A_{shadowi,j}$ is the wake shadow area at turbine j caused by turbine i, and A_0 the rotor area of turbine i and $def_{i,j}$ the single wake deficit from turbine i on turbine j.

The calculation of the wake effects is performed online during the simulation using Matlab S-functions embedded in the wind-field model. The computed wake deficits at each sampling time is added to the mean wind speed at each turbine to realise the wake effects during the simulation.

3.2 Wind Turbine Model

The wind turbine model is reported in Section 2.1.

In addition, the turbulent wind speeds generated for the wind-field model in Section 3.1 have a sampling rate of 1Hz. However the sampling rate for the wind turbine model needs to be higher, e.g. 40Hz or 100Hz, when discretized. Hence, the higher frequency contributions to the turbulent wind speeds must be added locally at each wind turbine as discussed below.

3.2.1 High frequency turbulence data points

The higher frequency components of the turbulent wind time series are generated using the Dryden spectrum and saved to be used later during online simulations. Dryden spectrum parameters are defined such that the time series approximate those generated by the Kaimal spectrum. The algorithm is described as follows.

Random high frequency data points are generated using Matlab normal distribution random generator "$normrnd(mue, sige)$" with the mean and standard deviation defined as

$$\mu = \frac{\sinh\left(a(t-t_i)\right) x\left(t_f\right) + \sinh\left(a\left(t_f - t\right)\right) x(t_i)}{\sinh(a(t_f - t_i)} \tag{20}$$

$$\delta^2 = 2\frac{\sinh\left(a(t-t_i)\right) \sinh\left(a\left(t_f - t\right)\right)}{\sinh(a(t_f - t_i)} b^2/(2a) \tag{21}$$

where, $a = \frac{1.14\bar{v}}{L}, b = T_i\sqrt{2a}, L = 200m$. T_i denotes turbulence intensity, t_i the initial time, t_f the final time of the interval between the two points

An algorithm is developed in Matlab to generate the high frequency components for the turbulence according to defined mean and variance for a random normal process. Initially, the first two consecutive points in the time series at $t = 0, t = 1$ are considered as the initial and final points in the algorithm to generate high frequency data points in between. The corresponding values for $x(t_i = 0), x(t_f = 1)$ are obtained from the generated turbulence time series data from the previous stage. The distance between t_i, t_f is divided by $N + 1$ for computing N additional data points between t_i, t_f. The algorithm then generates a statistically correlated random data points between these two points, using mean and standard deviation calculated using equations (20) and (21).

In the next step, the algorithm considers the new generated point as the initial point and generates another random data point between this point and the final point. This process is repeated to obtain as many data points as required between the two initial points in the low frequency time series. Once the required data points between the first initial and final data points at $t = 0, t = 1$ are generated, the algorithm repeats the process for the next time step $t = 1, t = 2$. Figure 19 illustrates the generated high frequency data points (red dots) between data points in the low frequency time series.

Figures 20 and 21 illustrate the time series and spectrum of the low frequency turbulence generated by the wind-field generator code and the added high frequency component generated by the Dryden spectrum. It shows that the high frequency components are randomly.

To minimise the calculation time further, a large integration time step can be selected for the wind-field model, e.g. 1s, while choosing a small integration step time for the wind turbine model, e.g. 0.025s. Moreover, they are pre-calculated to be read later during simulations. Although the wind farm model in Simulink is initially continuous in time, some sections of it are discretised in order to improve the simulation speed by converting those sections into C++ code to be used by Simulink s-functions.

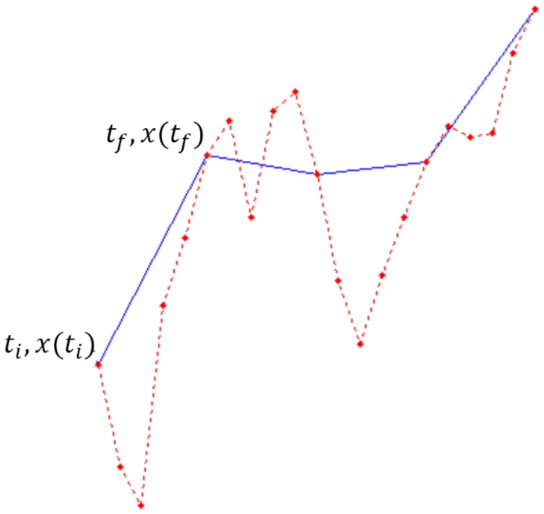


Figure 19: Dryden interpolation [1].

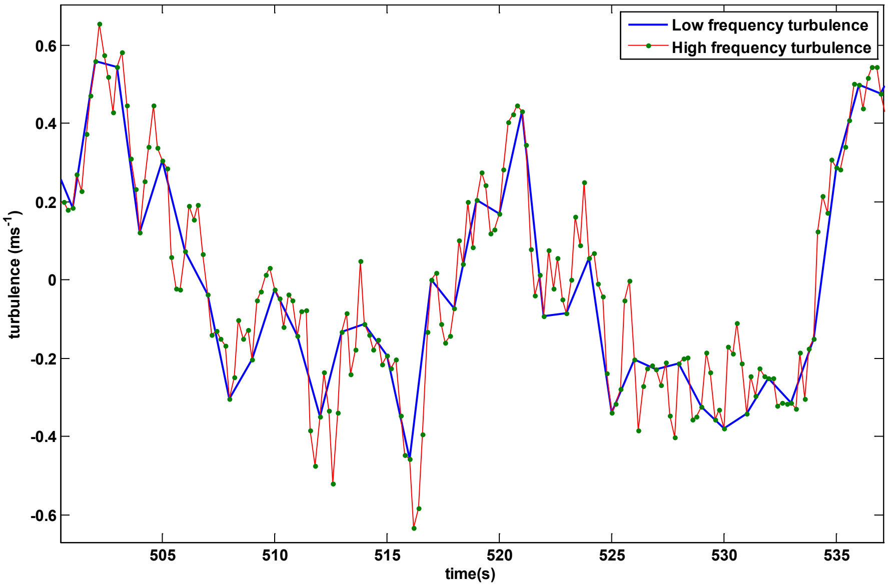


Figure 20: High frequency turbulence [1].

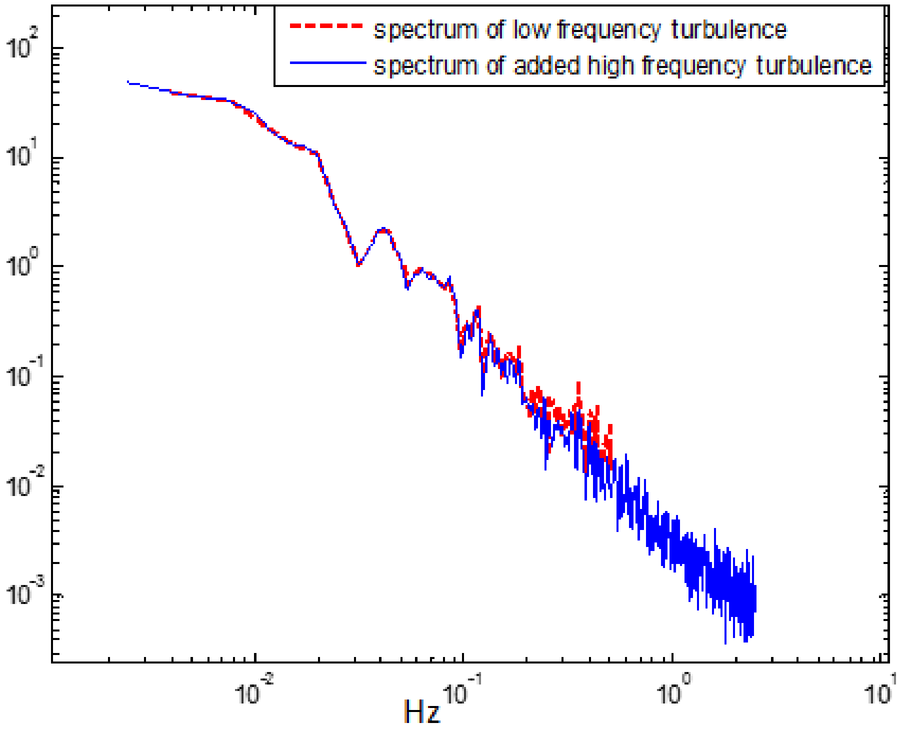


Figure 21: Spectra of the low and high frequency turbulence time series [1].

The wind turbine dynamics is modelled entirely as a continuous system. It thus is a stiff system for which a variable step solver is essential, for example, a suitable Simulink solver is ode23, an implicit variable step algorithm. The full envelope controller is implemented as a discrete section, hence an alternative solver, the ode14x, which is a fixed step solver that also works efficiently, could also be used. The discretization of the wind turbine model has not been done to date but is left to future work.

The wind farm control algorithm is developed in C/C++ and Matlab scripts, enabling the design of multiple control loops at different sampling rate.

3.2.2 Generating Effective Wind Speed

The wind stochastically varies with time and continuously interacts with the rotor [7]. The effective wind speed is wind speed averaged over the rotor area so that the power spectrum of aerodynamic torque remains intact. In this paper, it is derived by filtering the point wind speed through the filter introduced in [7]. The effective wind speeds are required to simulate the Simulink models.

3.3 Wind Farm Control Algorithm

This report is concerned with wind farm modelling only, but in order to allow simulations to be carried out the wind farm controller introduced in [1] is utilised.

3.4 Wind Farm Modelling Tool

The modelling tool developed in Matlab allows a larger number of wind turbines to be automatically built following user defined parameters, including wind turbine position, mean wind speed, turbulence intensity, sampling time and wind-field length and width.

The Matlab script loads the wind farm template model from the library and saves the created model in the current directory. Afterwards, the script copies the wind turbine model, for as many as required by the user, into the wind farm model template, and configures the subsystem components corresponding to the number of wind turbines. Finally, it connects the subsystems input/output ports automatically to complete the wind farm model development. The Matlab script then loads the wind farm model and its parameters to start the simulation.

4 Simulation of the Wind Farm

A wind farm model with 50 turbines placed in 5 rows parallel to the wind direction with 800m distance between them is developed and used throughout this section for testing the performance of the wind farm model described in Section 3. The wind farm controller from Section 3.3 is used here to curtail the wind farm power output by 5% of the available total power output for the purpose of simulating the wind farm, not the wind farm controller.

Figures 22, 23 and 24 demonstrate the operation of the 250MW wind farm at a mean wind speed of 8 m/s for 4000s. At low wind speeds, the PAC starts a recovery process, and the incremented power, Δp, of the turbine is reduced to zero. Since the available turbines (i.e., the ones with their PAC outside recovery mode) dictates the level of possible adjustment to the power output at any given time, in order to dispatch a sensible level of power set-point between the available turbines, the farm dispatch unit adjusts the total farm reference power to be less than the total available adjustment power level of the farm. The wind farm controller identifies wind turbines operating at low wind speeds and subsequently sets their power set-point adjustment level to zero, allowing them to recover.

This low speed recovery process is evident in the thrust coefficient of wind turbine 1 (WT1) depicted in Figure 22. As demonstrated in the figure, at low wind speeds, around 1300s and 2000s, the PAC goes into the recovery mode; that is, Δp is reduced to zero. Thrust coefficient also becomes what it would be under the normal operating conditions, i.e. without the PAC.

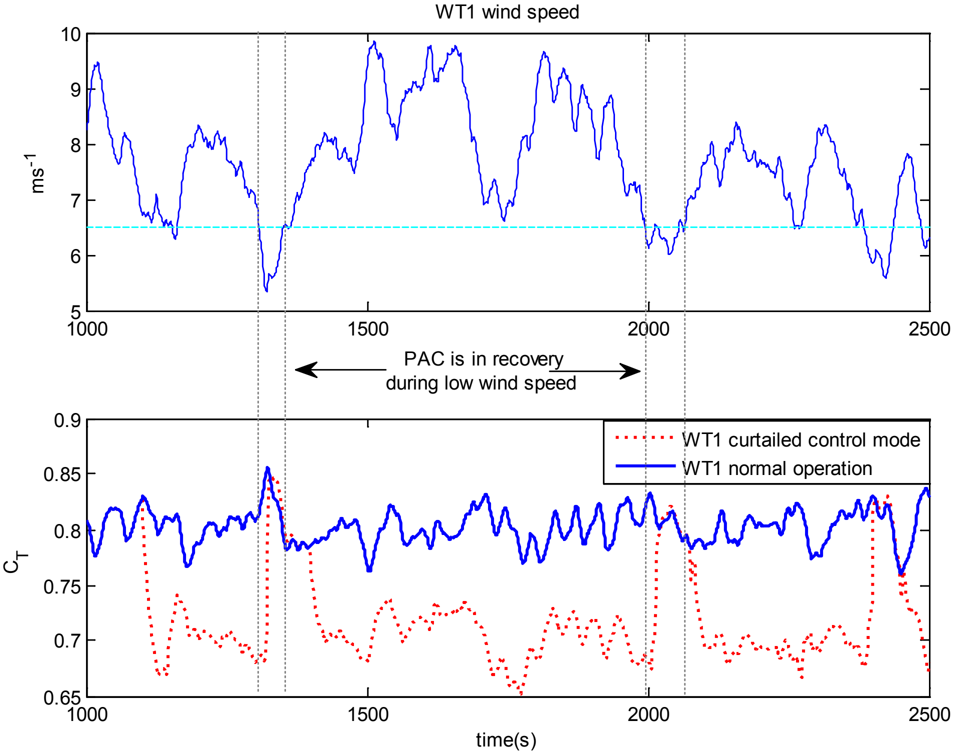


Figure 22: Low wind speed PAC recovery [1].

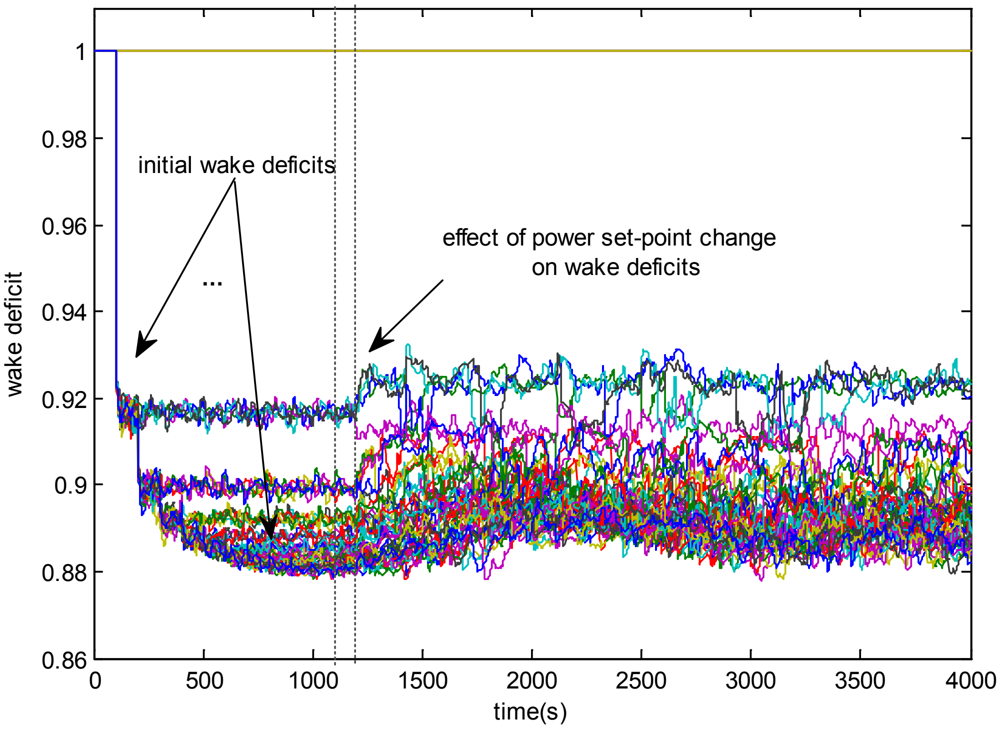


Figure 23: Wake deficits after power curtailment at 1100s [1].

Figure 23 illustrates velocity deficit at each turbine. It is in the transient state until 900s, and from 1100s the wind farm power output is curtailed by 5% of the available power output, i.e. the wind speed dictated output that would arise with no adjustment. Since all turbines are operating in the green traffic light region (see Figure 8), all the turbines participate in the power curtailment control operation. This contribution is evident at 1200s.

Figure 24 shows another scenario in which the total power output with and without the curtailment and the difference are plotted. Some turbines experience low wind speeds, and their PAC remains in the recovery mode. Since the PACs of the turbines experiencing low wind speeds are in the recovery mode, not contributing to the curtailment operation, the difference in green in the figure is small at times.

In another situation, the power curtailment level achieved immediately after the power adjustment request at 1000s as depicted in Figure 25. It is shown that the wind farm power output can be adjusted in a quick and safe operating manner.

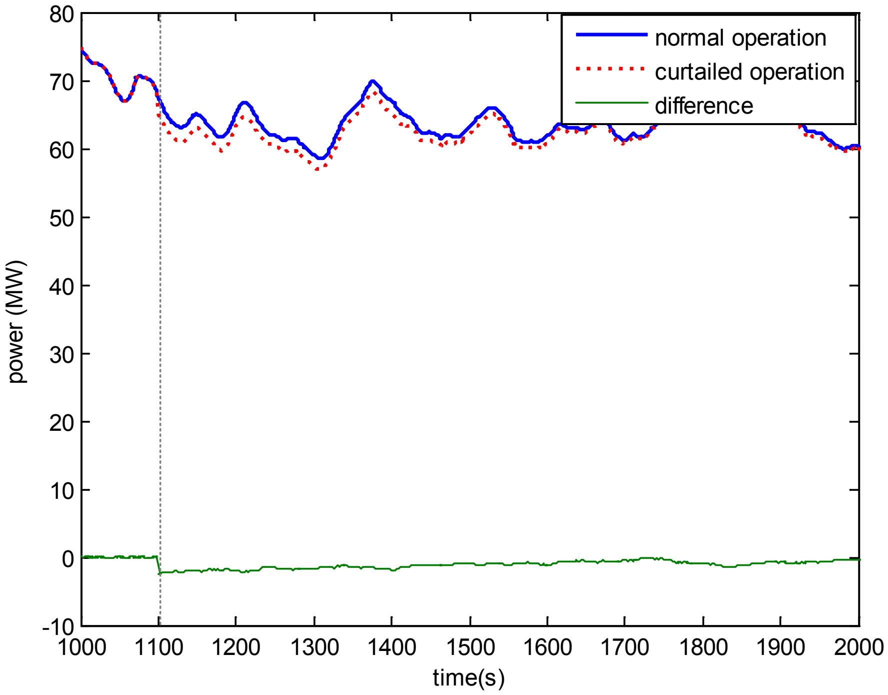


Figure 24: Total power output before and after curtailment of 250MW wind farm in below rated 8 m/s wind speed [1].

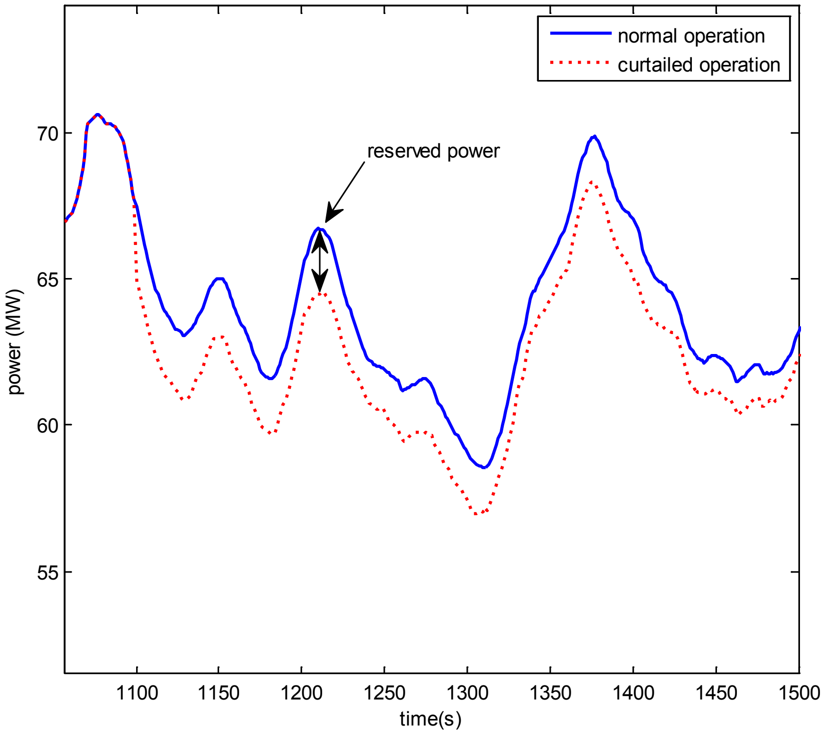


Figure 25: Reserved power at curtailed control mode [1].

Table 1: Simulation time for wind farms of different sizes

Number of turbines	Simulation time (s)	Real time elapsed (s)
10	600	356.25
20	600	779.33
50	600	3598.12

The time it takes to run the wind farm models of 10, 20 and 50 turbines for 600s is presented in Table 1. It takes approximately 1 hour in real time to run the wind farm model of 50 turbines for 600s in simulation time. Considering the complexity of each turbine and the wind field model, in addition to the incorporation of the full envelope controller and a PAC dedicated to each turbine and the wind farm controller, it seems to be adequate. In future, the simulation time will be improved further by discretising the turbine model (i.e. dynamics) and implementing it in C++.

5 Summary

The procedure of developing a wind farm model for fast simulation and wind farm controller design is presented. A wind farm model construction tool is also developed to build wind farm models with a large number of wind turbines automatically. It accepts user defined parameters and builds the corresponding wind farm model.

The wind farm model includes a wind-field model which is fast enough for simulation but still detailed enough for wind turbine load analysis. The wind-field model also includes the effect of wake. The wind farm model also contains a wind farm control algorithm that is developed for coordinating control of the wind turbines in the farm. However, the wind farm controller design is beyond the scope of this document and is not discussed in detail.

Each turbine model in the wind farm model includes not only the turbine dynamics, but also the full envelope controller (which is important as they have significant impact on the operation of the wind turbines and farms) and a PAC, which is used for adjusting the turbine's power output according to a power set-point change request from the wind farm control algorithm in a quick and safe operating manner.

The wind farm model is divided in continuous and discrete sections where appropriate to improve the simulation time. The model runs on the basis of multiple sampling rate simulation that allows for flexible wind farm controller design. To date, only the full envelope controller and the PAC are implemented in C++ (and some parts of the wind farm controller), but in future, the turbine dynamics will also be discretised and implemented in C++ to improve the simulation time further.

The wind farm model is tested for various operating conditions and control scenarios. The results suggest that the wind farm model is suitable for wind farm control design.

Bibliography

[1] S. Poushpas, “Wind farm simulation modelling and control,” University of Strathclyde, Ph.D. thesis, Glasgow, 2016.

[2] A. Stock, “Augmented Control for Flexible Operation of Wind Turbines,” University of Strathclyde, Ph.D. thesis, Glasgow, 2014.

[3] A. Chatzopoulos, “Full Envelope Wind Turbine Controller Design for Power Regulation and Tower Load Reduction,” University of Strathclyde, Ph.D. thesis, Glasgow, 2011.

[4] W. Leithead, D. Leith, F. Hardan and H. Markou, “Global gain-scheduling control for variable speedwind turbines,” in *European Wind Energy Conference*, 1999.

[5] P. S. Veers, “Three-Dimensional wind Simulation,” 1988.

[6] S. Frandsen, R. Barthelmie, S. Pryor, O. Rathmann and S. Larsen, “Analytical modelling of wind speed deficit in large offshore wind farms,” *Wind Energy,* pp. 39-53, 2006.

[7] W. Leithead, “Effective wind speed models for simple wind turbine simulations,” in *Proceedings of 14th British Wind Energy Association (BWEA) Conference*, Nottingham, 1992.

[8] S.-h. Hur and W. E. Leithead, “Adjustment of wind farm power output through flexible turbine operation using wind farm control,” *Wind Energy,* pp. 1-20, 2015.

StrathFarm Wake Modelling Study: Part 1

Lindsey Amos

Wind and Marine Energy CDT, University of Strathclyde

E-mail: `lindsey.amos@strath.ac.uk`

June 2019

Contents

1 Introduction **2**

2 The Frandsen Wake Model **3**

3 Medium-Fidelity Wake Models **6**

4 WFSim in StrathFarm **7**

5 StrathFarm Wake Models Compared to High-Fidelity Data **8**

5.1 Row of 5 turbines, below rated 8

5.2 Row of 5 turbines, above rated 9

5.3 Other Simulation Results 11

5.4 Comment on Results 11

6 Study Part 2 and Future Work **12**

Appendix A PALM Comparison : Row 5WT Below Rated **16**

Appendix B PALM Comparison : Row 5WT Above Rated **16**

Appendix C PALM Comparison : Row 5WT Above Rated **17**

Appendix D PALM Comparison : Row 5WT Below Rated, 15 degrees **19**

Appendix E PALM Comparison : 6WT Grid Below Rated **21**

1. Introduction

This report covers the findings of the first part of the study into the wake model used within StrathFarm - the University of Strathclyde control-oriented dynamic wind farm simulation tool. StrathFarm will be used for the design and analysis of wind farm control algorithms, aimed at optimising for power and loads, by influencing wake interactions in the wind farm via changing the power set-point of each turbine (axial induction control).

Wake modelling in wind farm simulation is a trade-off between computational expense (simulation time) and accuracy. It is important to have confidence in the wake model. Axial induction control produces counteracting effects, since when thrust force is reduced on a turbine, the downwind near wake velocity increases but recovery decreases (effectively decreasing the velocity on the far wake).

Wind farm simulation wake models range from computationally expensive, high-fidelity CFD models based on the 3D governing equations of fluid flow, to computationally cheap heuristic, parametric or engineering wake models, where the idea is to capture the most dominant wake characteristics. These engineering models are 2D and predict a linearly expanding wake with a velocity deficit that only depends on the distance behind the rotor, they also employ fairly crude methods of combining multiple wakes (and it is known that under multiple wake conditions the equivalent loads due to different wakes are not additive). In between are so-called medium fidelity wake models, which trade off the accuracy of high-fidelity models with the computational complexity of low-fidelity models. These are in general based on simplified versions of the NavierStokes equations.

Note: a full literature and background review on wind farm simulation wake models has been omitted from this report for the sake of brevity.

2. The Frandsen Wake Model

The popular kinematic engineering wake model of Frandsen [1] is widely used. It is used by SimWindFarm [2], and the pre-cursor to StrathFarm presented in [3]. Analysis of the performance of the Frandsen engineering model at Horn's Rev wind farm presented in [4] is cited by [3] and illustrated in Figure 1.

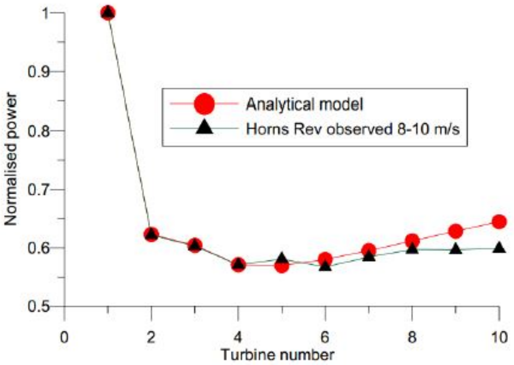


Figure 1: Frandsen Engineering Model comparison to data from Horn's Rev [4]. 8-10m/s wind speeds.

A more informative plot regarding Horn's Rev, since there are different inter-turbine spacings within the wind farm is shown in Figure 2. Also note that Horn's Rev contains 2MW machines.

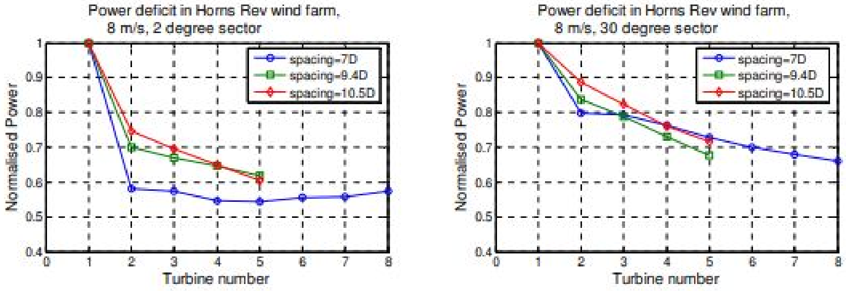


Figure 4: Power deficit inside Horns Rev wind farm for V=8±0.5 m/s inflow for different spacing.

Figure 2: Frandsen Engineering Model comparison to data from Horn's Rev [5]

The Frandsen engineering wake model contains a wake expansion factor, α. In both SimWindFarm [2] and the initial StrathFarm implementation the value of α is a constant, $\alpha = 0.5$, apparently cited to Frandsen.

Polster's comprehensive comparison of analytical wind turbine wake models [6] finds $\alpha \approx 10k_w$, where k_w is the Jensen wake decay constant. They state $k_w \approx 0.1$, however most other literature finds k_w to be a function of surface roughness and hence the ambient turbulence intensity. Pena and Rathmann [7] describe k_w as the ratio between the free friction velocity and the free hub height wind speed; through the analysis of these values Pena et al. [8] finds $k_w \approx 0.4 * T_i$. The literature therefore suggests setting $\alpha \approx 4 * TI$, e.g. for $T_i = 5\%$, $\alpha = 0.2$.

In [1] α is defined as:

$$\alpha = \beta^{k/2}\left[(1 + 2\alpha_{(noj)}s)^k - 1\right]s^{-1} \tag{1}$$

Where $\beta = \frac{1+\sqrt{1-C_T}}{2\sqrt{1-C_T}}$, $\alpha_{(noj)} = 0.05$, $k = 2$ and s is the downwind distance from wake generating wind turbine in rotor diameters. Thus for $s = 7$ for a 7D spacing case $\alpha = 0.27\beta$.

We therefore have the following cases for α:

- Alpha constant: $\alpha = 0.5$, as in [2],[3].
- Alpha linear with TI: $\alpha = 4T_i$, as suggested in the literature.
- α calculated from β: $\alpha = 0.27\beta$, [1] (for 7D inter-turbine spacing).

StrathFarm simulations were run for a row of ten 5MW wind turbines, with 7D spacing between the turbines in the row. For a turbulence intensity $T_i = 5\%$, $\alpha = 4T_i = 0.2$. By running the time series simulation for long enough for the wakes to fully develop along the row and then extracting a further 600s of data, the results for the normalised mean power are shown in Figure 3, where the results for $\alpha = 0.2$ are in the region of what is expected based on the Horn's Rev data.

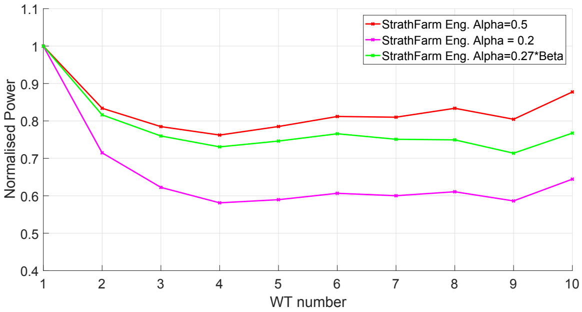


Figure 3: Normalised Power for a Row of 10 wind turbines, 7D spacing, 8m/s 5% : Comparison of Frandsen Wake Model in StrathFarm varying Alpha

At a turbulence intensity of 11%, $\alpha = 4T_i = 0.44$; which is much closer to the constant value of $\alpha = 0.5$ used by [2],[3]. From the StrathFarm results presented in Figure 4 for a wind field with a mean wind speed 9m/s and turbulene intensity of 11%, all the suggested values of α give a conservative wake deficit of ≈ 0.8 at the first downwind turbine and a trend showing toward nearly full wake recovery along the row. It is only

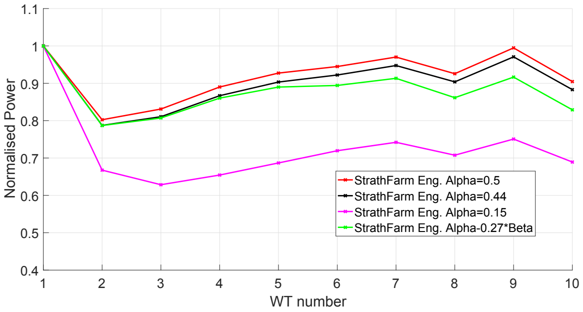


Figure 4: Normalised Power for a Row of 10 wind turbines, 7D spacing, 9m/s 11% : Comparison of Frandsen Wake Model in StrathFarm varying Alpha

with a value of $\alpha = 0.15$ that the deficits appear more in line with the values in the literature.

Alongside the interest in studying and developing an alternative wake model for wind farm simulation, these results indicate for the Frandsen wake model:

- Using a fixed value of $\alpha = 0.5$ ([2] and [3]) seems crude, since it represents a wake expansion factor. Using this fixed value gives conservative results, i.e. the wake effects are not as great as the literature would suggest they should be.
- The linear relationship $\alpha = 4T_i$ provides more realistic results than $\alpha = 0.5$, particularly for the $T_i = 5\%$ case. However, the $T_i = 11\%$ case demonstrates that this method results in lower wake effects and at higher turbulence the kind of level of power deficit suggested by the literature is achieved by a lower value of $\alpha = 0.15$.

It is not clear from these results whether the higher α values are correct and the results are a reasonable representation of the wake in higher turbulence conditions. If this is not the case, a method of tuning α beyond any of those above is required. A more accurate, even if more computationally expensive wake model could perhaps be used to do this.

Note: the choice of a conservative wake model using $\alpha = 0.5$ may well have been justified for initial StrathFarm simulation and analysis. If one is concerned with the potential increase in wind farm power output available by optimising the power set-points of the turbines, it would be sensible to use a wake model which acts conservatively, so to tend not to overestimate the potential of the wind farm controller.

3. Medium-Fidelity Wake Models

Medium-fidelity wake models are essentially a trade-off between accuracy and computational expense, achieved by simplifying the 3D Navier-Stokes equations. Most medium-fidelity models are 2D. NREL's 2D Dynamic Wake Meandering (DWM) [9] model reduces computational effort through approximations to the Navier-Stokes equations from assumptions made regarding the thin shear layer. Annoni and Seiler [10] apply time averaging to the NavierStokes equations, resulting in the 2-D RANS equations. The number of states is then reduced by employing a state reduction technique. Models based on the discretised 2-D Navier-Stokes equations [11], [12], [13], have also proved interesting; and initially the author took to implementing a model similar to [11] in StrathFarm. However, these models do not contain a turbulence model that allows for wake recovery.

Neglecting vertical effects entirely has been shown to effect the accuracy of the wake model result. Atmospheric stability effects are also important, increasingly so as we consider large offshore wind turbines, and a wind farm simulation model should be developed to incorporate this.

WFSim [14] is a medium-fidelity wake model, recently developed at DTU for online control. It is based on the 2D Navier-Stokes equations but a modification is made to partially correct for the neglected vertical dimension. The model is solved using the finite volume method with a hybrid differencing scheme over a staggered grid; and contains a heuristic turbulence model (Prandtl's mixing length).

WFSim has previously been presented in the literature [14] [15], but the implementation in StrathFarm was novel as it is used as a wake deficit model only, and in this setting it is coupled to wind turbine models which can provide a good estimate of key loads. In the literature WFSim had been validated against high-fidelity data using an input time series of the turbine thrust coefficient and axial induction factor, also from the high-fidelity data.

The motivation to explore the implementation of a medium-fidelity wake model in StrathFarm should be clear, and WFSim meets many of the desired requirements. However, the computational expense of even a medium-fidelity model may be intractable for large farm simulations, and may not be necessary if it's results can be used to improve and tune the engineering wake model sufficiently.

4. WFSim in StrathFarm

The WFSim model was available to the author as a standalone MATLAB code with example data. The input data to the standalone model included time series for the wind turbine coefficients of thurst and axial induction factors.

For the StrathFarm implementation of WFSim the code acts as the wake deficit model, applying a wake deficit to the mean wind speed seen by each wind turbine at each time step. The coefficient of thrust and axial induction factor from each wind turbine model is input to the WFSim wake model. The WFSim code was implemented C++ (utilising the Armadillo [16] [17] matrix library with assiciated solvers) and compiled as a dynamic link library. A linker s-fucntion allows the WFSim wake model library to be used in the StrathFarm simulink model.

The selection of grid size for the WFSim model was made to allow the simulations to be run at practical computational expense (at least faster than real-time); the grid size used in most cases is 42m, i.e. a third of the wind turbine rotor diameter.

For the row of ten wind turbines with 7D spacing, the results of using WFSim as the wake model in StrathFarm can be compared to the Frandsen model. For the 8m/s, 5% case in Figure 5 the WFSim model results match quite closely the Frandsen model results with $\alpha = 0.2$. For the 9m/s,11% case the WFSim results presented are in line with the reduced value of $\alpha = 0.15$.

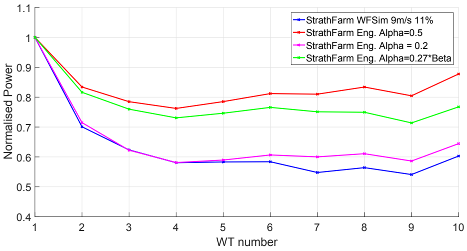


Figure 5: Row of 10 wind turbines, 7D spacing, 8m/s 5% : Comparison of Wake Models

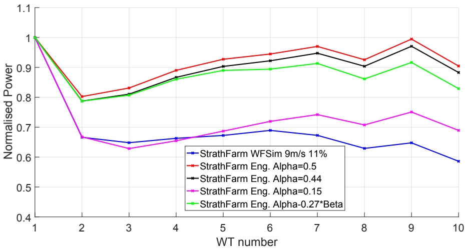


Figure 6: Row of 10 wind turbines, 7D spacing, 9m/s 11% : Comparison of Wake Models

5. StrathFarm Wake Models Compared to High-Fidelity Data

One of the requirements for this study was to obtain high-fidelity wind farm simulation data to compare the StrathFarm wake models to. Initially, since the author had access and familiarity with ANSYS Windmodeller, time was spent gaining access to University of Strathclyde's HPC cluster and evaluating the requirements to run a series of Windmodeller simulations to generate the data for this purpose; this could still be processed to a full study on the cluster by purchase of the required core hours. The use of Windmodeller as a comparison also seemed sensible when considering the simplified 2D RANS medium-fidelity wake models.

However, WFSim has been compared to high-fidelity data from initially SOWFA [18], and more recently with PALM [19] [20]. The author of WFSim was kind enough to share these and some further PALM datasets so that a comparison to WFSim in StrathFarm could be undertaken. The results of those comparisons are the subject of the remainder of this section of this report. Both below and above rated examples are considered.

Alongside the presentation of the comparison of WFSim in StrathFarm to the high-fidelity data, results using the Frandsen engineering model in StrathFarm are also shown, both with $\alpha = 0.5$ and with α manually changed to closer reflect the PALM values.

5.1. Row of 5 turbines, below rated

The first result to consider is for a row of five wind turbines, with 7D inter-turbine spacing. For a below rated mean wind speed of 8m/s and a turbulence intensity of 6%, the WFSim wake model captures the downwind mean velocity decrease and turbulence intensity increase, Figures E2 and E3 respectively. The PALM and StrathFarm wind fields have non-identical ambient turbulence, which mostly explain the difference seen for the most downwind turbines.

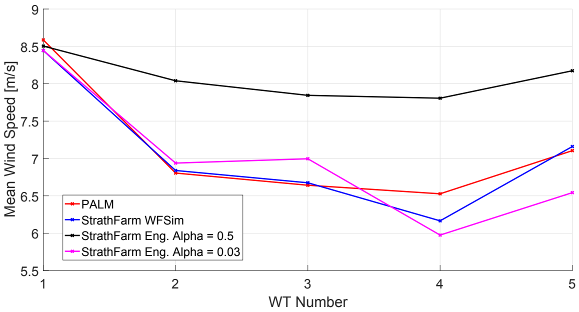


Figure 7: 5WT 7D Spacing Below Rated: Mean wind speed at each wind turbine

Considering the results for the Frandsen Engineering model, a value of $\alpha = 0.03$ gives a closer comparison to the PALM data than the constant $\alpha = 0.5$.

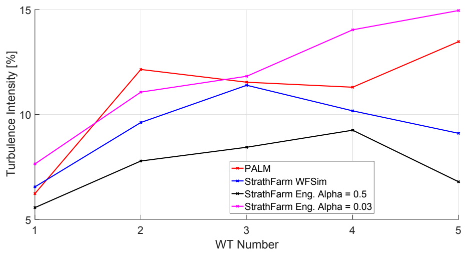


Figure 8: 5WT 7D Spacing Below Rated: Turbulence intensity at each wind turbine

Plots of the wind speed time series for each wind turbine, used to generate the statistics presented here are given in Appendix A, alongside the simulation times and other information.

5.2. Row of 5 turbines, above rated

In above rated wind conditions, a row of five 5MW turbines is considered, with both a 7D and 5D inter-turbine spacing. The closer inter-turbine spacing produces a stronger aerodynamic coupling between the turbines in the row. For the 7D case a mean wind speed of 14m/s and turbulence intensity of 5% is simulated; for the 5D case a mean wind speed of 13m/s and turbulence intensity of 9%.

For the 7D case, the resulting mean wind speed and turbulence intensity for 600s of fully-waked simulation data is presented in Figures 9 and **??** respectively. As can be seen, the WFSim model represents the wake profile well, albeit with lower levels of turbulence. The Frandsen Engineering model with $\alpha = 0.5$ underpredicts the wakes, a value of $\alpha = 0.06$ produces a closer comparison.

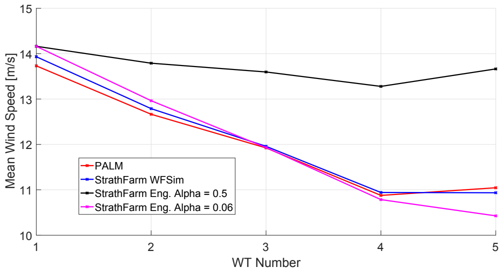


Figure 9: 5WT 7D Spacing Above Rated: Mean wind speed at each wind turbine

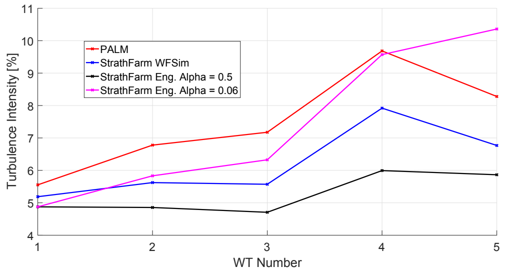


Figure 10: 5WT 7D Spacing Above Rated: Turbulence intensity at each wind turbine

Plots of the wind speed time series for each wind turbine, used to generate the statistics presented here are given in Appendix C, alongside the simulation times and other information.

Similar results are presented for the 5D case in Figures 11 and **??**, where $\alpha = 0.05$ produces a closer comparison to the PALM data. Plots of the wind speed time series for each wind turbine, used to generate the statistics presented here are given in **??**, alongside the simulation times and other information.

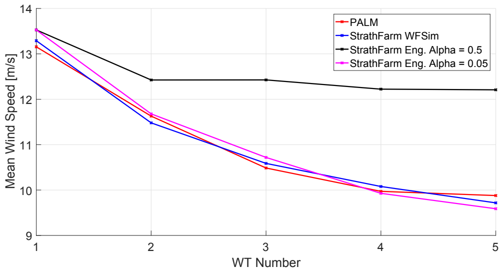


Figure 11: 5WT 7D Spacing Above Rated: Mean wind speed at each wind turbine

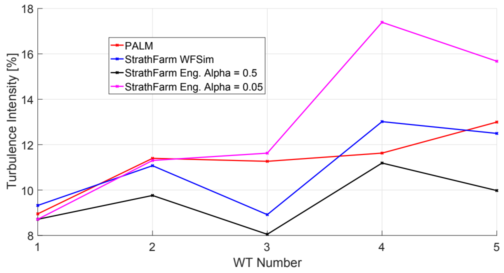


Figure 12: 5WT 7D Spacing Above Rated: Turbulence intensity at each wind turbine

Like the below rated case, both above rated cases show the Frandsen engineering model with $\alpha = 0.5$ to underestimate the wake interaction. Again, the value of α can be reduced to find a result which closer compares to the PALM and WFSim results. Taking the results for these values of α and comparing the PALM and StrathFarm data for the 7D and 5D cases, does show the expected result for reducing the inter-turbine spacing i.e. a stronger wake interaction, as shown in Figures 13 and 14.

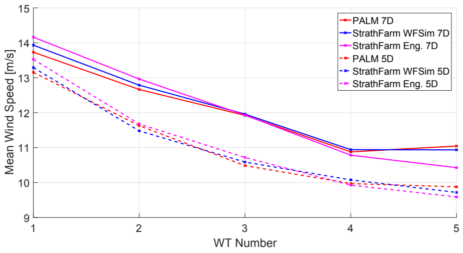


Figure 13: 5WT Above Rated, Compare 7D and 5D Spacing: Mean wind speed at each wind turbine

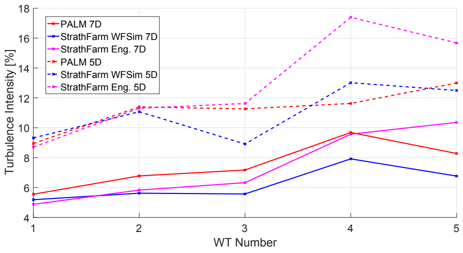


Figure 14: 5WT Above Rated, Compare 7D and 5D Spacing: Turbulence intensity at each wind turbine

5.3. Other Simulation Results

Some other PALM datasets were provided, information regarding each of these is given below:

- **Row of 5WT, 7D spacing, below rated, 15 degrees wind direction**
  It proved difficult to create a StrathFarm wind field with ambient conditions that could represent the PALM data. However, a WFSim appears to be able to represent this scenario more realistically than the Frandsen Engineering model. The results for this scenario are summarised in Appendix D.

- **Row of 5WT, 7D spacing, below rated, low surface roughness**
  Here the mean wind speed at the first wind turbine in the PALM data is under 7m/s and the results show a drop to around 5m/s along the row. Given that the cut-in wind speed for the 5MW turbine is 4m/s and the controller's used by the wind turbine's in PALM and StrathFarm will not be the same, the models cannot be reasonably compared.

- **2x3 Grid, 5D row spacing, 3D spacing between rows, below rated**
  The original PALM dataset, results in Appendix E. The WFSim model compares well to the PALM data considering the high turbulence seen for WTs 3 and 4 in the PALM data which appear part of the ambient turbulence. The Frandsen Engineering model with $\alpha = 0.5$ performs quite poorly in comparison. α for this scenario has not been manually tuned.

5.4. Comment on Results

The results of the comparison of StrathFarm wake models to high-fidelity datasets indicate that the Frandsen low-fidelity engineering wake model with constant $\alpha = 0.5$ fails to capture the wake behaviour accurately. The medium-fidelity wake model WFSim in StrathFarm results compare well to high-fidelity data, and by tuning α the engineering wake model can be much improved.

The WFSim model has parameters that require tuning, the results in this section suggest that it is possible to do this using high-fidelity data. Further data is required to determine a reliable relationship between the tunable parameters and the input wind, although the author is quite confident that this can be done reasonably easily. The fact that there is only a single wake expansion factor, α, may still make the tuning and use of the Frandsen wake model limited, however, in it's improved form (with a tuned α) it is likely to be satisfactory for many wind farm simulation tasks.

The option to have a medium-fidelity wake model in StrathFarm is useful, not only to have a model which is hopefully more accurate than the engineering wake model, but also the inclusion of the vertical dimension (albeit simplified and included as a correction), not only increases accuracy but makes future considerations of atmospheric stability a possibility.

Referring back to Figures 5 and 6 which looked at the row of 10 turbines simulated below rated in StrathFarm. While α could be tuned so that the results from the two models matched closely for the first 5 turbines in the row, the results diverge further along the row, with the engineering wake model demonstrating more wake recovery. It would be good to confirm with high-fidelity data which model is capturing the expected behaviour more accurately.

6. Study Part 2 and Future Work

The purpose of this report was to present the results of the initial study of two wake models in the StrathFarm simulation tool, a low-fidelity engineering model and a medium-fidelity model using WFSim. WFSim has previously been presented in the literature, but the implementation in StrathFarm was novel as it is used as a wake deficit model only, and in this setting it is coupled to wind turbine models which can provide a good estimate of key loads. In the literature WFSim had been validated against high-fidelity data using an input time series of the turbine thrust coefficient and axial induction factor, also from the high-fidelity data.

The aim of this wake modelling study is to meet the requirement for the StrathFarm simulation tool to have a wake model or models which are well-understood and which the user can be confident in over a range of practical wind conditions.

The contents of this report have shown that with a constant wake expansion factor, the popular kinematic engineering wake model of Frandsen is probably insufficient. The medium-fidelity wake model using WFSim appears to provide more accurate results and could be used in simulations where the accuracy of the wake model is more important, or to tune the engineering model.

In order to further the study and gain more confidence in the models it would be particularly useful to obtain more high-fidelity data. This could be used to further determine and verify the WFSim tuning to input wind conditions, and indeed the tuning of the engineering wake model. More below rated than above rated scenarios would be preferable. High-fidelity data from a row of ten wind turbines could also help clarify what the downwind behvaiour past wind turbine 5 should be, as both StrathFarm models produce different levels of wake recovery. More data in grid formations would also prove useful for ensuring lateral effects are being modelled sufficiently accurately.

Beyond the tuning of the available parameters in the wake models it is also noted that the StrathFarm wake model at present only acts to provide a deficit to the mean wind speed seen by a wind turbine (thus modelling the wake behaviour of decrease in mean and increase in turbulence intensity). In StrathFarm the local rotor effective wind field components do not have any wake effect applied to them, this leaves an area to be studied, with potential improvement to the way wakes are incorporated in the StrathFarm model.

The abstract submitted for the WESC presentation and paper in June 2019 left some flexibility as to what is presented. The focus is to present a study of the wind turbine loads from simulations using WFSim in StrathFarm; I would like to see whether and how turbulence calculated from the wind speed estimator in each PAC [21] compares to the fatigue loads from the wind turbines (eventually I want to consider whether we can use turbulence as a proxy for fatigue within a wind farm control algorithm that has access to the wind speed estimator values). To do these simulations I will need tunings of WFSim I am confident in, based on comparison to high-fidelity data. The presentation could also include a comparison to the results achieved with the Frandsen engineering model and/or some simulations in StrathFarm where some wind farm control or curtailment is undertaken.

References

[1] S. Frandsen, R. Barthelmie, S. Pryor, O. Rathmann, S. Larsen, J. Hojstrup, and M. Thogersen, "Analytical Modelling of Wind Speed Deficit in Large Offshore Wind Farms," tech. rep., Wind Energy; 9:39-53, 2006.

[2] V. Spudic, M. Baotic, M. Jelavic, and N. Peric, "Hierarchical wind farm control for power/load optimization," tech. rep., Proc. Conf. Sci. Making Torque from Wind, pp. 681-692, 2010.

[3] S. Poushpas, "Wind Farm Simulation Modelling and Control," tech. rep., PhD Thesis, University of Strathclyde, 2016.

[4] M. Mechali, R. Barthelmie, S. Frandsen(2), L. Jensen, and P.-E. Rethore, "Wake effects at Horns Rev and their influence on energy production," tech. rep., European Wind Energy Conference, 1, pp. 1020, 2006.

[5] R. Barthelmie, O. Rathmann, S. Frandsen, K. Hansen, E. Politis, J. Prospathopoulos, K. Rados, D. Cabezn, W. Schlez, J. Phillips, A. Neubert, J. Schepers, and S. van der Pijl, "Modelling and measurements of wakes in large wind farms," tech. rep., Journal of Physics Conference Series 75 012049, 2007.

[6] M. Polster, "Comprehensive comparison of analytical wind turbine wake models with wind tunnel measurements and wake model application on performance modelling of a downstream turbine," tech. rep., Masters Thesis, NTNU, 2017.

[7] A. Pena and O. Rathmann, "Atmospheric stability-dependent infinite wind-farmmodels and the wake-decay coefficient," tech. rep., Wind Energy 17:12691285, 2014.

[8] A. Pena, P.-E. Rethore, and M. P. van der Laan, "On the application of the Jensen wake model using a turbulence-dependent wake decay coefficient: the Sexbierum case," tech. rep., Wind Energy 19:763-776, 2016.

[9] "NWTC Information Portal (DWM)." https://nwtc.nrel.gov/DWM, Last modified 06-April-2015.

[10] J. Annoni and P. Seiler, "A low-order model for wind farm control," tech. rep., American Control Conference, 2015.

[11] M. Soleimanzadeh, R. Wisniewski, and A. Brand, "State-space representation of the wind flow model in wind farms," tech. rep., Wind Energy 17:627-639, 2014.

[12] M. V. S. Boersma, J. W. van Wingerden and M. Khn, "Quasi linear parameter varying modeling for wind farm control using the 2D Navier Stokes equations," tech. rep., American Control Conference, Boston, 2016.

[13] M. V. B. M. D. S. Boersma, P. M. O. Gebraad, "A control-oriented dynamic wind farm flow model: WFSim," tech. rep., TORQUE, 2016.

[14] S. Boersma, B. Doekemeijer, M. Vali, J. Meyers, and J.-W. v. Wingerden, "A control-oriented dynamic wind farm model: WFSim," tech. rep., Wind Energy Science, 3, 7595, 2018.

[15] B. Doekemeijer, S. Boersma, L. Pao, T. Knudsen, and J.-W. v. Wingerden, "Online model calibration for a simplified LES model in pursuit of real-time closed-loop wind farm control," tech. rep., Wind Energy Science, 3, 749-765, 2018.

[16] C. Sanderson and R. Curtin, "Armadillo: a template-based C++ library for linear algebra," tech. rep., Journal of Open Source Software, Vol. 1, 2016.

[17] C. Sanderson and R. Curtin, "A user friendly hybrid sparse matrix class in C++," tech. rep., Lecture Notes in Computer Science, Vol. 10931, 2018.

[18] J. M. M. Churchfield, S. Lee and P. Moriarty, "A numerical study of the effects of atmospheric and wake turbulence on wind turbine dynamics," tech. rep., Journal of Turbulence, 2012.

[19] B. Witha, S. Gerald, and D. Heinemann, "Advanced turbine parameterizations in offshore LES wake simulations," tech. rep., International Symposium on Computational Wind Engineering, 2014.

[20] B. Maronga, M. Gryschka, R. Heinze, F. Hoffmann, F. Kanani-Shring, M. Keck, K. Ketelsen, M. O. Letzel, M. Shring, and S. Raasch, "The Parallelized Large-Eddy Simulation Model (PALM)

version 4.0 for atmospheric and oceanic flows: model formulation, recent developments, and future perspectives," tech. rep., Geosci. Model Dev., 8, 25152551, 2015.

[21] A. Stock, "Augmented Control for Flexible Operation of Wind Turbines," tech. rep., PhD Thesis, University of Strathclyde, 2015.

Appendix A. PALM Comparison : Row 5WT Below Rated

Simulation times for generating 1300s of data (use last 600s of fully waked simulation for statistics):

- WFSim StrathFarm = 365.8s;
- Eng StrathFarm = 29.5s.

z_0 = 0.05, forcescale = 1.0, lm_slope = 0.1.

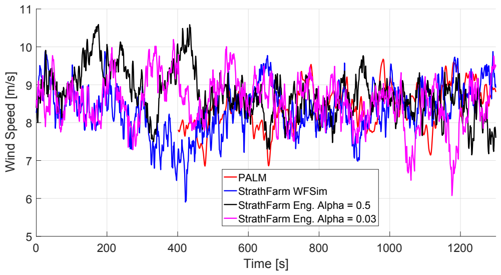


Figure A1: Wind Speed Time Series WT1

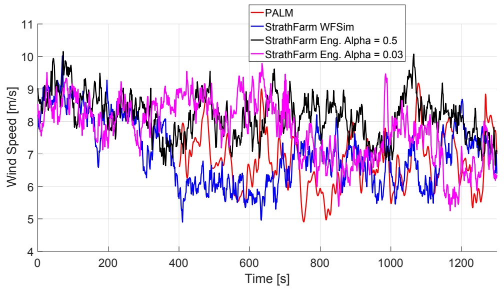


Figure A2: Wind Speed Time Series WT2

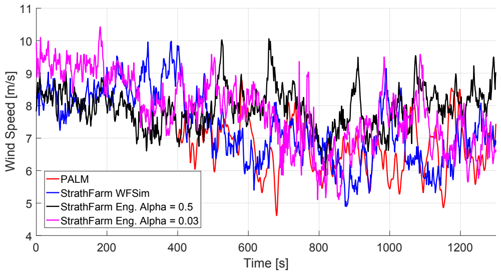


Figure A3: Wind Speed Time Series WT3

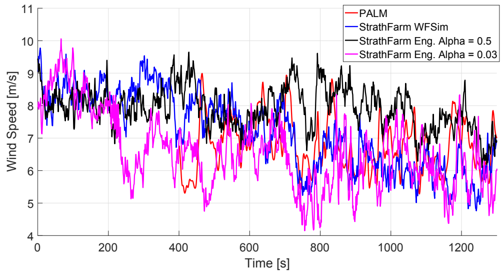


Figure A4: Wind Speed Time Series WT4

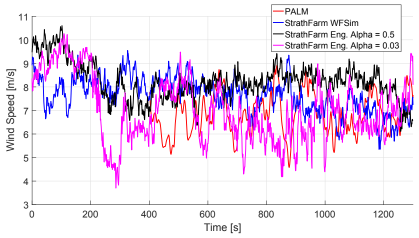


Figure A5: Wind Speed Time Series WT5

Appendix B. PALM Comparison : Row 5WT Above Rated

StrathFarm imulation times for generating 1000s of data (use last 600s of fully waked simulation for statistics):

- WFSim StrathFarm = 294.36s;
- Eng StrathFarm = 35.39s.

7D Spacing, $z_0 = 0.05$, forcescale = 1.25, lm_slope = 0.18.

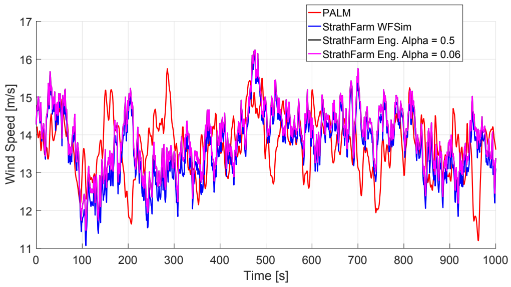


Figure B1: Wind Speed Time Series WT1

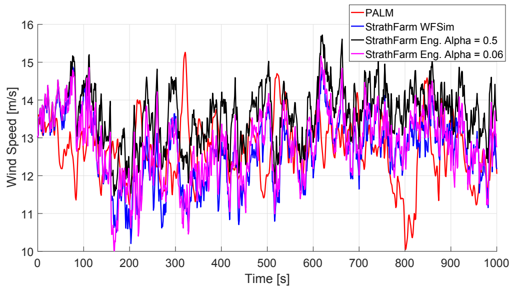


Figure B2: Wind Speed Time Series WT2

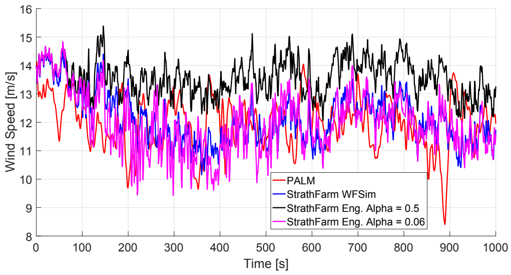


Figure B3: Wind Speed Time Series WT3

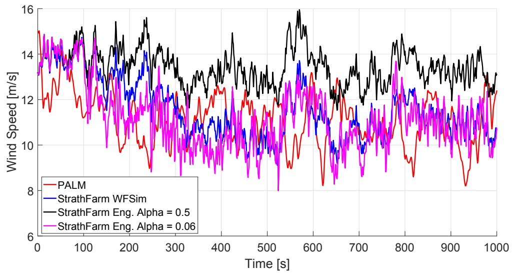


Figure B4: Wind Speed Time Series WT4

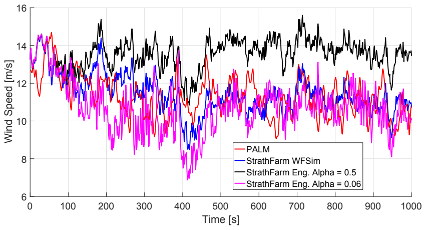


Figure B5: Wind Speed Time Series WT5

Appendix C. PALM Comparison : Row 5WT Above Rated

Simulation times for generating 1000s of data (use last 600s of fully waked simulation for statistics):

- WFSim StrathFarm = 215.91s;
- Eng StrathFarm = 51.2s.

5D Spacing, $z_0 = 0.05$, forcescale = 0.8, lm_slope = 0.15.

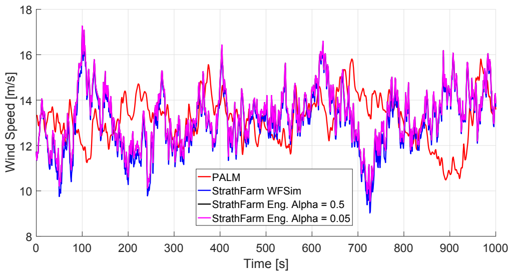


Figure C1: Wind Speed Time Series WT1

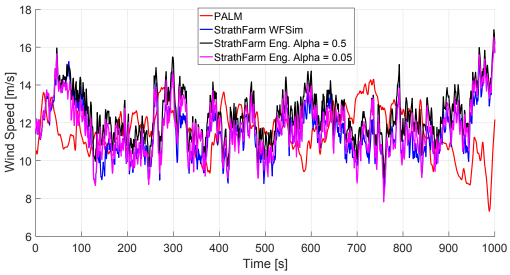


Figure C2: Wind Speed Time Series WT2

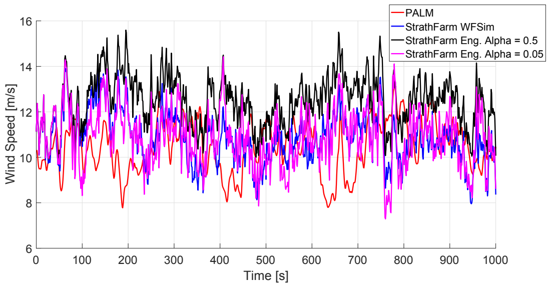


Figure C3: Wind Speed Time Series WT3

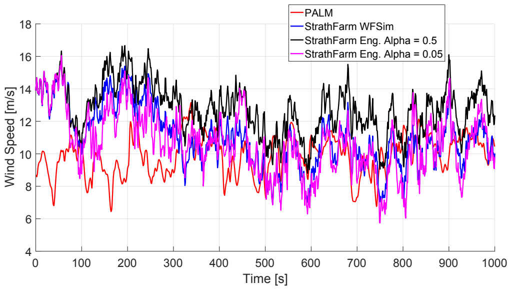


Figure C4: Wind Speed Time Series WT4

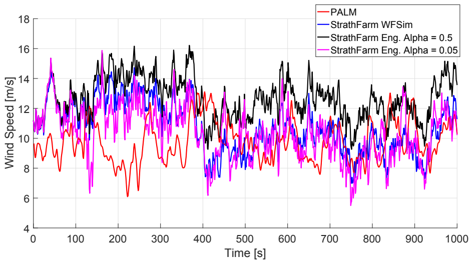


Figure C5: Wind Speed Time Series WT5

Appendix D. PALM Comparison : Row 5WT Below Rated, 15 degrees

A wind direction of 15 degrees can be considered as applying the layout change shown in the Figures below:

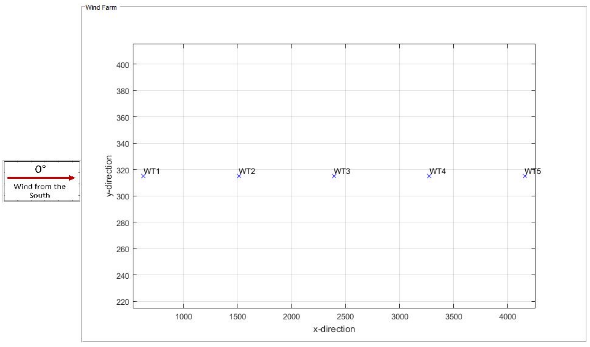


Figure D1: Turbine Positions Relative to Wind Direction: 0 degrees

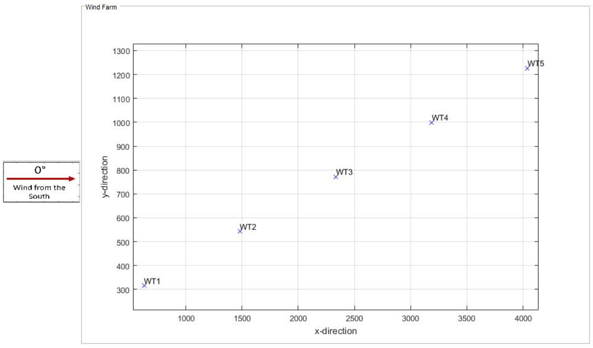


Figure D2: Turbine Positions Relative to Wind Direction: 15 degrees

Simulation times for generating 1800s of data (use last 600s of fully waked simulation for statistics):

- WFSim StrathFarm = 2254s;
- Eng StrathFarm = 63.7s.

7D Spacing, $z_0 = 0.05$. 15 degrees wind direction.

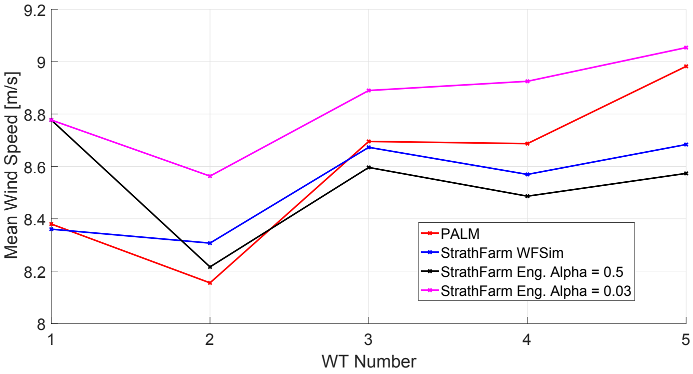


Figure D3: 5WT 5D Spacing Below Rated: Mean wind speed at each wind turbine

Note it was difficult to get a wind field in StrathFarm that could represent this kind of scenario - since the wind speed in the PALM data is highest at the most downwind wind turbine, although the turbulence is fairly constant. The WFSim forcescale was increased to achieve the WFSim results (increasing the force effect of each turbine also results in a reduced wind speed at the first turbine). The Frandsen engineering model applied to the same wind field thus gives a too high wind speed for the first wind turbine as it does not affect the ambient conditions of it and it is set high to give the correct

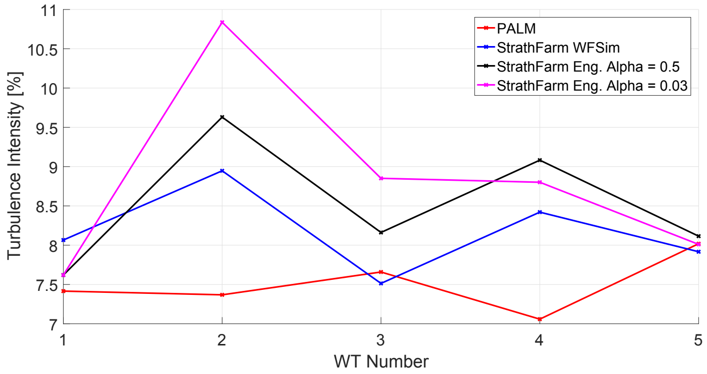


Figure D4: 5WT 7D Spacing Below Rated: Turbulence intensity at each wind turbine

result when reduced by the WFSim model. The results for $\alpha = 0.5$ and $\alpha = 0.03$ are still shown.

Appendix E. PALM Comparison : 6WT Grid Below Rated

In this scenario 6 WTs are positioned in 2 rows of 3 wind turbines, with a spacing of 5D along each row, and an inter-row spacing of 3D. A below rated wind speed is simulated with a mean wind speed of 8m/s and a turbulence intensity of 5%. The Frandsen Engineering model is only tested with $\alpha = 0.5$.

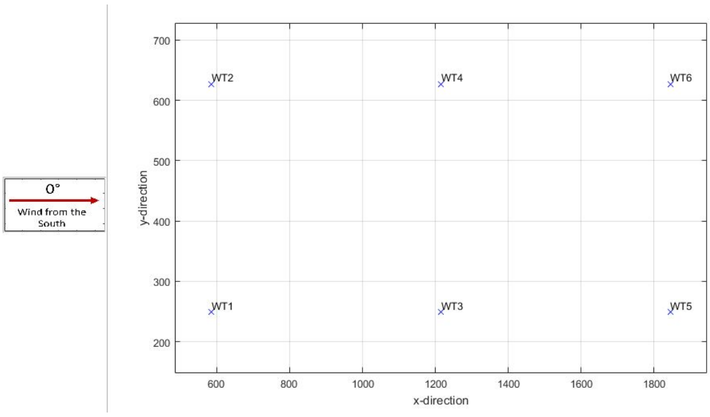


Figure E1: The layout for the 6WT PALM comparison

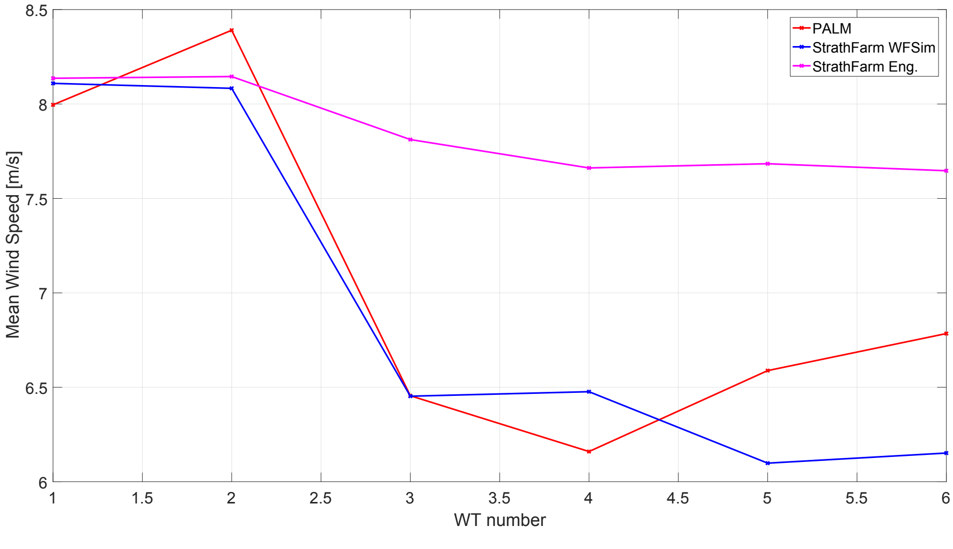


Figure E2: 6WT 5D Spacing Below Rated: Mean wind speed at each wind turbine

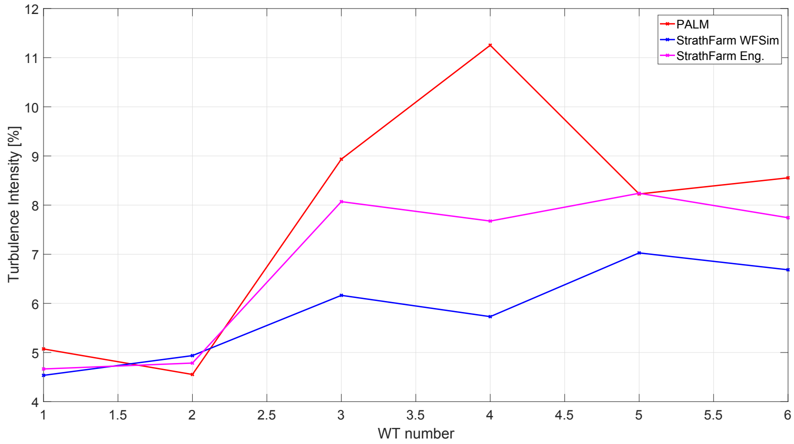
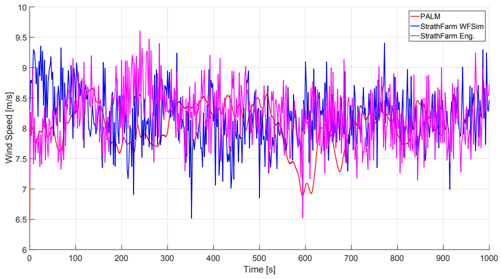


Figure E3: 6WT 5D Spacing Below Rated: Turbulence intensity at each wind turbine



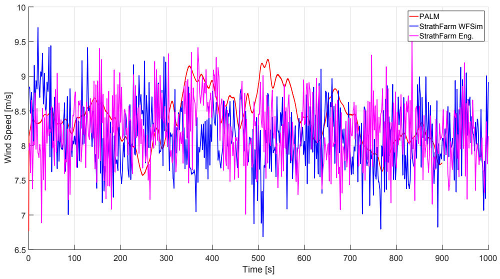
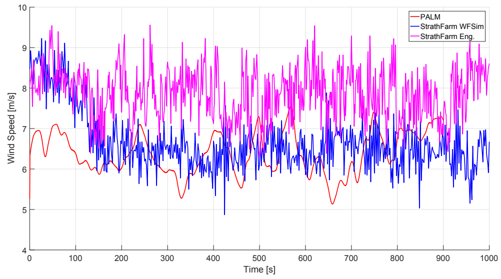


Figure E4: Wind Speed Time Series WT1

Figure E5: Wind Speed Time Series WT2



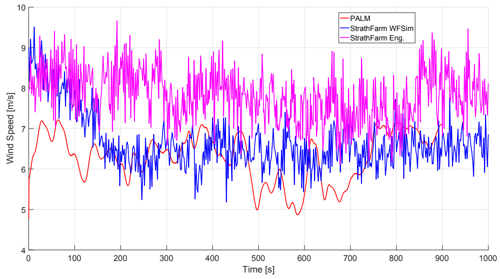
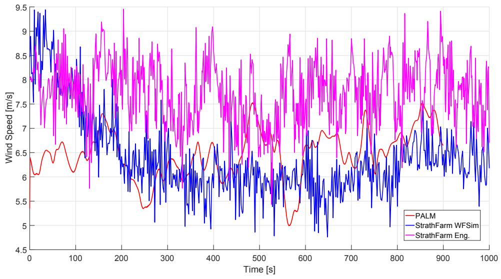


Figure E6: Wind Speed Time Series WT3

Figure E7: Wind Speed Time Series WT4



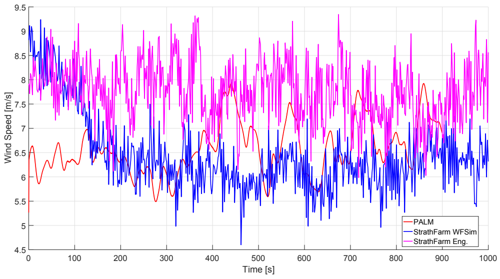


Figure E8: Wind Speed Time Series WT5

Figure E9: Wind Speed Time Series WT6

StrathFarm Wake Modelling Study: Part 1a

Lindsey Amos

Wind and Marine Energy CDT, University of Strathclyde

E-mail: `lindsey.amos@strath.ac.uk`

June 2019

Contents

1 Introduction **2**

2 Results **3**

2.1 0 degrees wind direction . . . 3

2.2 5 degrees wind direction . . . 4

2.3 10 degrees wind direction . . . 5

2.4 15 degrees wind direction . . . 6

2.5 Comparison of Results . . . 8

1. Introduction

This report is an addition to the report covering the findings of the first part of the study into the wake model used within StrathFarm - the University of Strathclyde control-oriented dynamic wind farm simulation tool.

This document contains the results of further comparison to high-fidelity PALM datasets. These new datasets are below rated simulations of 5 wind turbines in a row with 7D inter-turbine spacing. The Part 1 report contained the results for simulations at 0 and 15 degrees wind direction. These results are repeated here alongside new results for simulations at 5 and 10 degrees wind direction. Figure 1 shows the resulting turbine layout for the variation in wind direction.

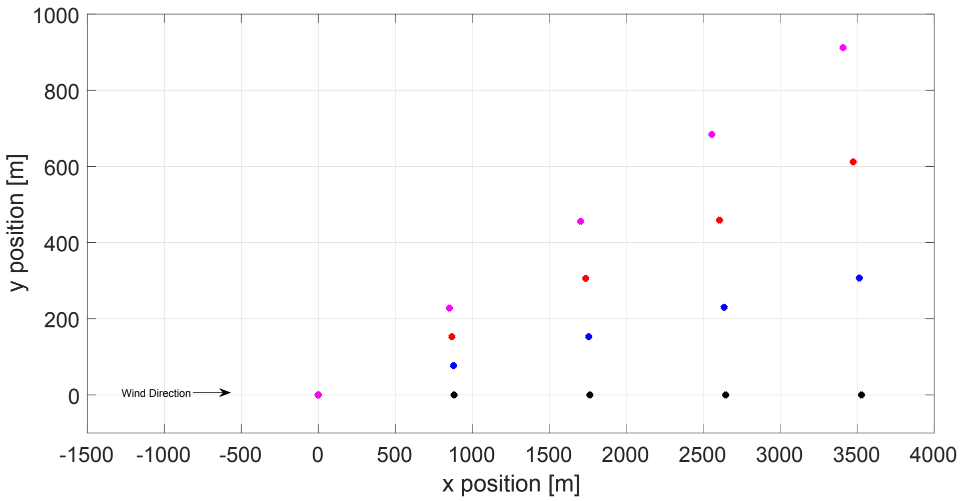


Figure 1: 5WT 7D Spacing Layouts Resulting from Change in Wind Direction

In the next section the results of comparing wake models in StrathFarm to the PALM datasets are given.

2. Results

2.1. 0 degrees wind direction

The first result is at 0 degrees wind direction. For a below rated mean wind speed of 8m/s and a turbulence intensity of 6%, the WFSim wake model captures the downwind mean velocity decrease and turbulence intensity increase, Figures 2 and 3 respectively. The PALM and StrathFarm wind fields have non-identical ambient turbulence, which mostly explain the difference seen for the most downwind turbines.

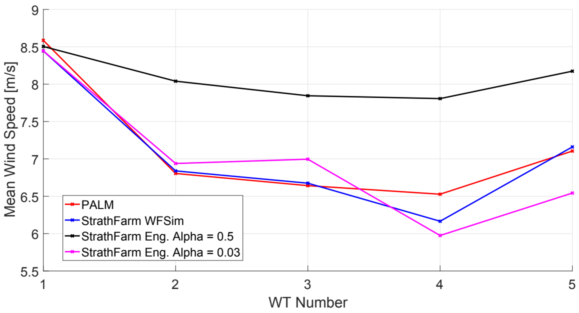


Figure 2: 5WT 7D Spacing Below Rated 0 degrees: Mean wind speed at each wind turbine

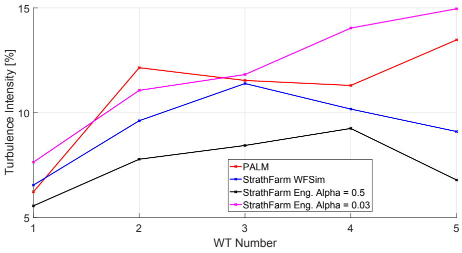


Figure 3: 5WT 7D Spacing Below Rated 0 degrees: Turbulence intensity at each wind turbine

Considering the results for the Frandsen Engineering model, a value of $\alpha = 0.03$ gives a closer comparison to the PALM data than the constant $\alpha = 0.5$.

2.2. 5 degrees wind direction

For a simulation with mean wind speed of 6.5m/s and a turbulence intensity of ≈6%, with a 5 degrees wind direction. There is a much reduced wake effect as shown in Figures 4 and 5. For the WFSim in StrathFarm simulation the overall mean deficit is 0.98 and the minimum is 0.96.

Simulation times for generating 1800s of data (use last 600s of fully waked simulation for statistics):

- WFSim StrathFarm = 1675.4s;
- Eng StrathFarm = 28.1s.

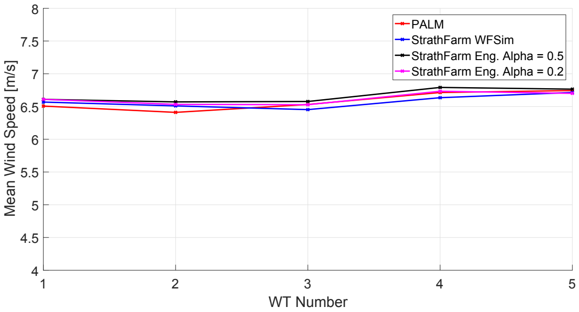


Figure 4: 5WT 7D Spacing Below Rated 5 degrees: Mean wind speed at each wind turbine

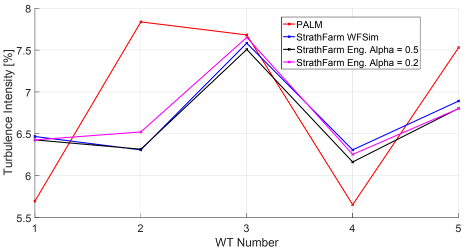


Figure 5: 5WT 7D Spacing Below Rated 5 degrees: Turbulence intensity at each wind turbine

2.3. 10 degrees wind direction

For a simulation with mean wind speed of 7.5m/s and a turbulence intensity of ≈5%, with a 10 degrees wind direction. There is a much reduced wake effect as shown in Figures 6 and 7. For the WFSim in StrathFarm simulation the overall mean deficit is 0.99 and the minimum is 0.98.

Simulation times for generating 1800s of data (use last 600s of fully waked simulation for statistics):

- WFSim StrathFarm = 1666.0s;
- Eng StrathFarm = 25.9s.

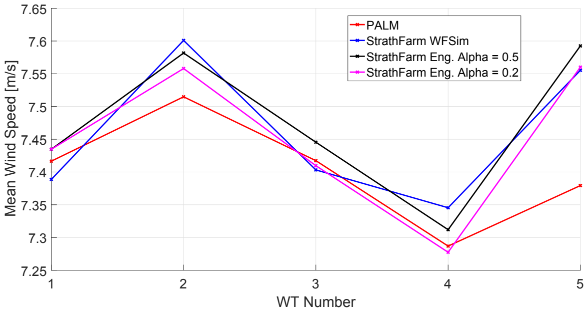


Figure 6: 5WT 7D Spacing Below Rated 10 degrees: Mean wind speed at each wind turbine

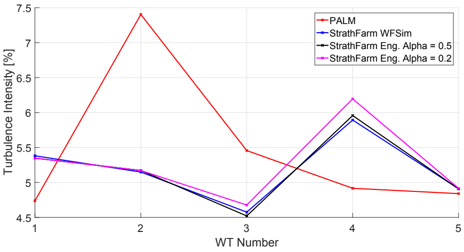


Figure 7: 5WT 7D Spacing Below Rated 10 degrees: Turbulence intensity at each wind turbine

2.4. 15 degrees wind direction

It proved difficult to create a StrathFarm wind field with ambient conditions that could represent the PALM data.

Simulation times for generating 1800s of data (use last 600s of fully waked simulation for statistics):

- WFSim StrathFarm = 2254s;
- Eng StrathFarm = 63.7s.

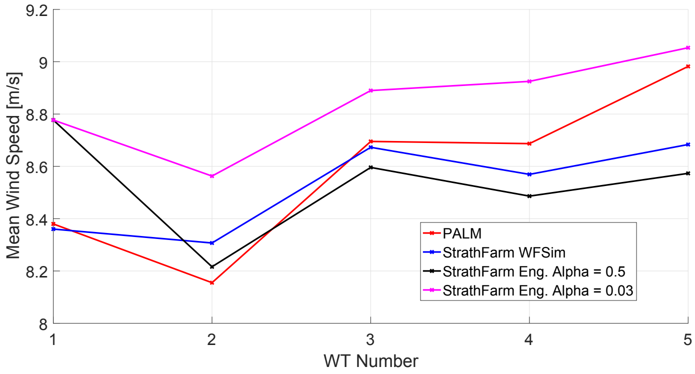


Figure 8: 5WT 7D Spacing Below Rated 15 degrees: Mean wind speed at each wind turbine

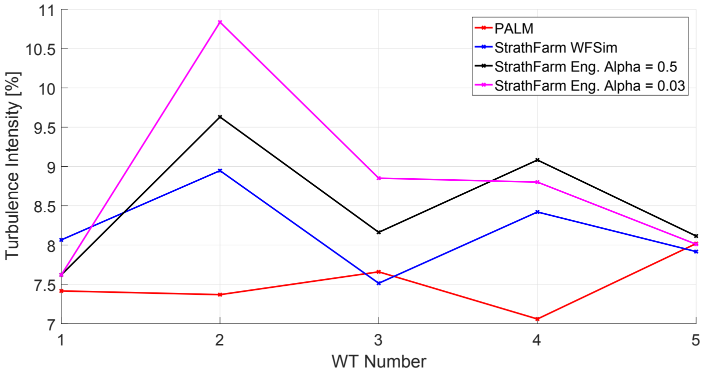


Figure 9: 5WT 7D Spacing Below Rated 15 degrees: Turbulence intensity at each wind turbine

Note it was difficult to get a wind field in StrathFarm that could represent this kind of scenario - since the wind speed in the PALM data is highest at the most downwind wind turbine, although the turbulence is fairly constant. The WFSim forcescale was increased to achieve the WFSim results (increasing the force effect of each turbine also results in a reduced wind speed at the first turbine). The Frandsen engineering model applied to the same wind field thus gives a too high wind speed for the first wind turbine as it does not affect the ambient conditions of it and it is set high to give the correct result when reduced by the WFSim model. The results for $\alpha = 0.5$ and $\alpha = 0.03$ are still shown.

The wake effect is not very great in this scenario so the results here are not very reliable and appear to be more fitting the ambient wind field than comparing the wake effects.

2.5. Comparison of Results

In Figures 10 and 11 the results for all 4 wind directions for the PALM and WFSim in StrathFarm data are given. The full wake effect for 0 degrees drops dramatically at 5 degrees. At 10 and 15 degrees the wake effect is insignificant.

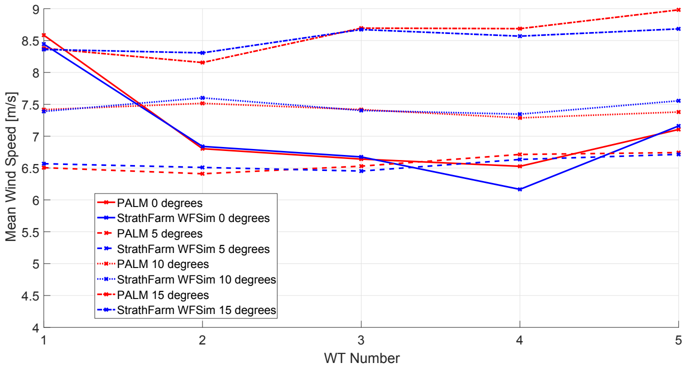


Figure 10: Changing Wind Direction: Mean wind speed at each wind turbine

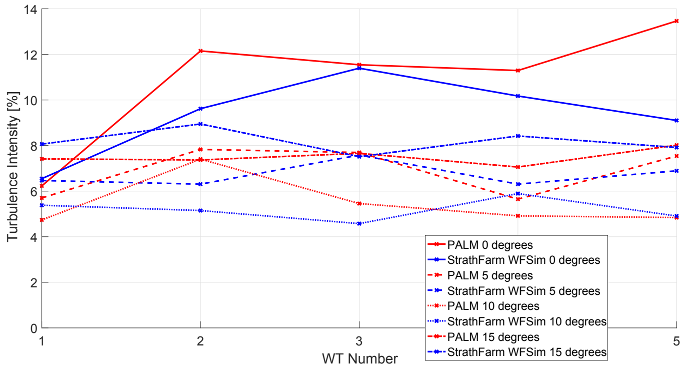


Figure 11: Changing Wind Direction: Turbulence intensity at each wind turbine

At 5 degrees wind direction the Frandsen engineering model performs just as well as the WFSim in StrathFarm model. This perhaps indicates that depending on the layout and conditions in StrathFarm the use of a medium-fidelity wake model is not necessary to improve accuracy.

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:

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

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

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

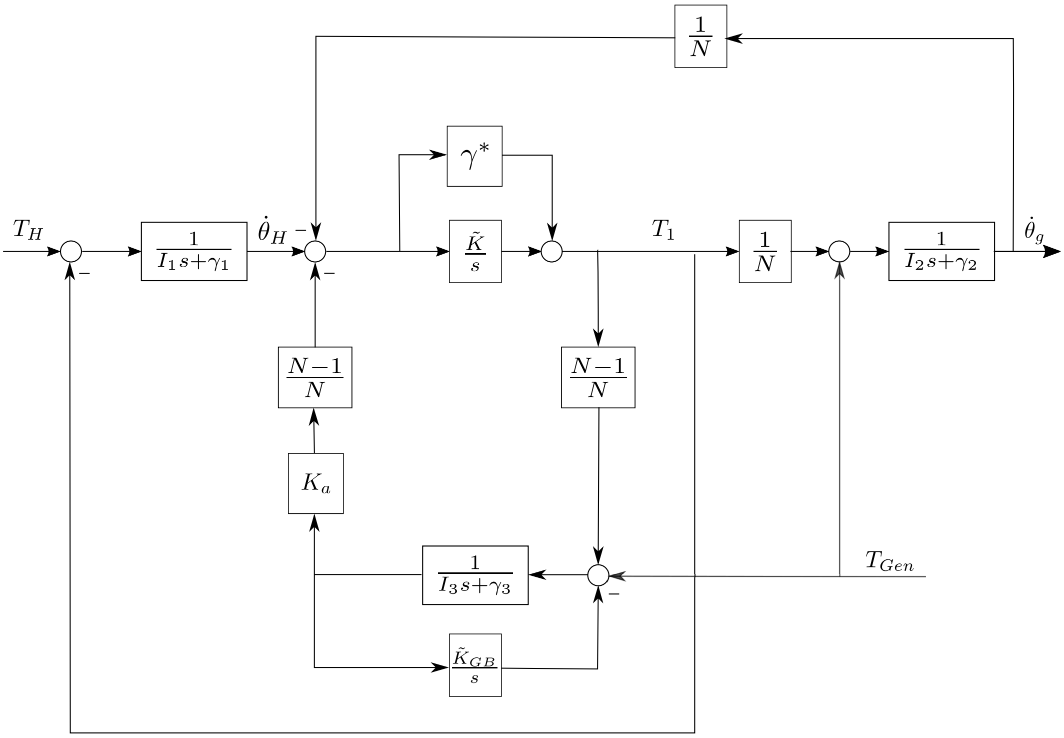


Figure 1: Drive-train module.

The hub torque dynamics is described by the following equation:

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

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

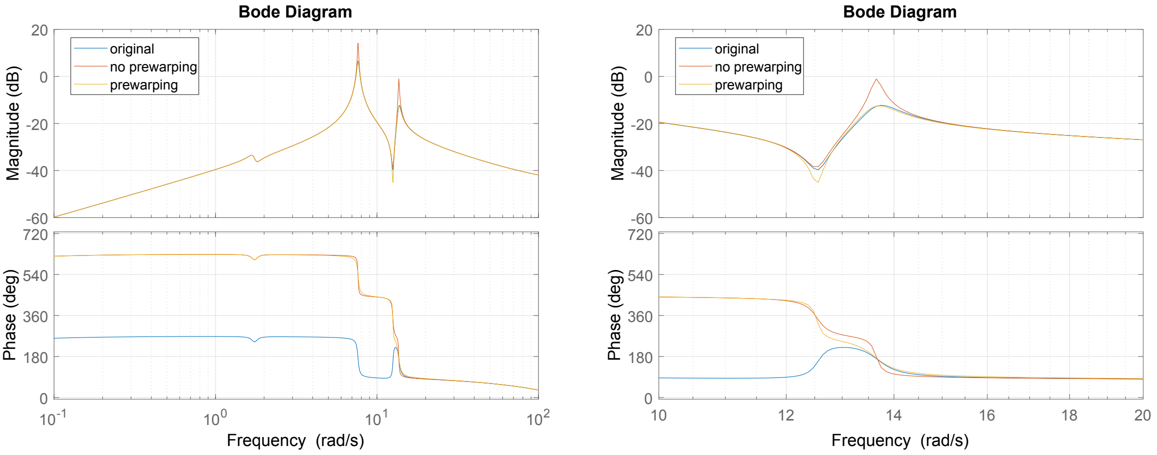


Figure 2: Discretisation and prewarping; the one on the RHS shows a zoomed version.

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

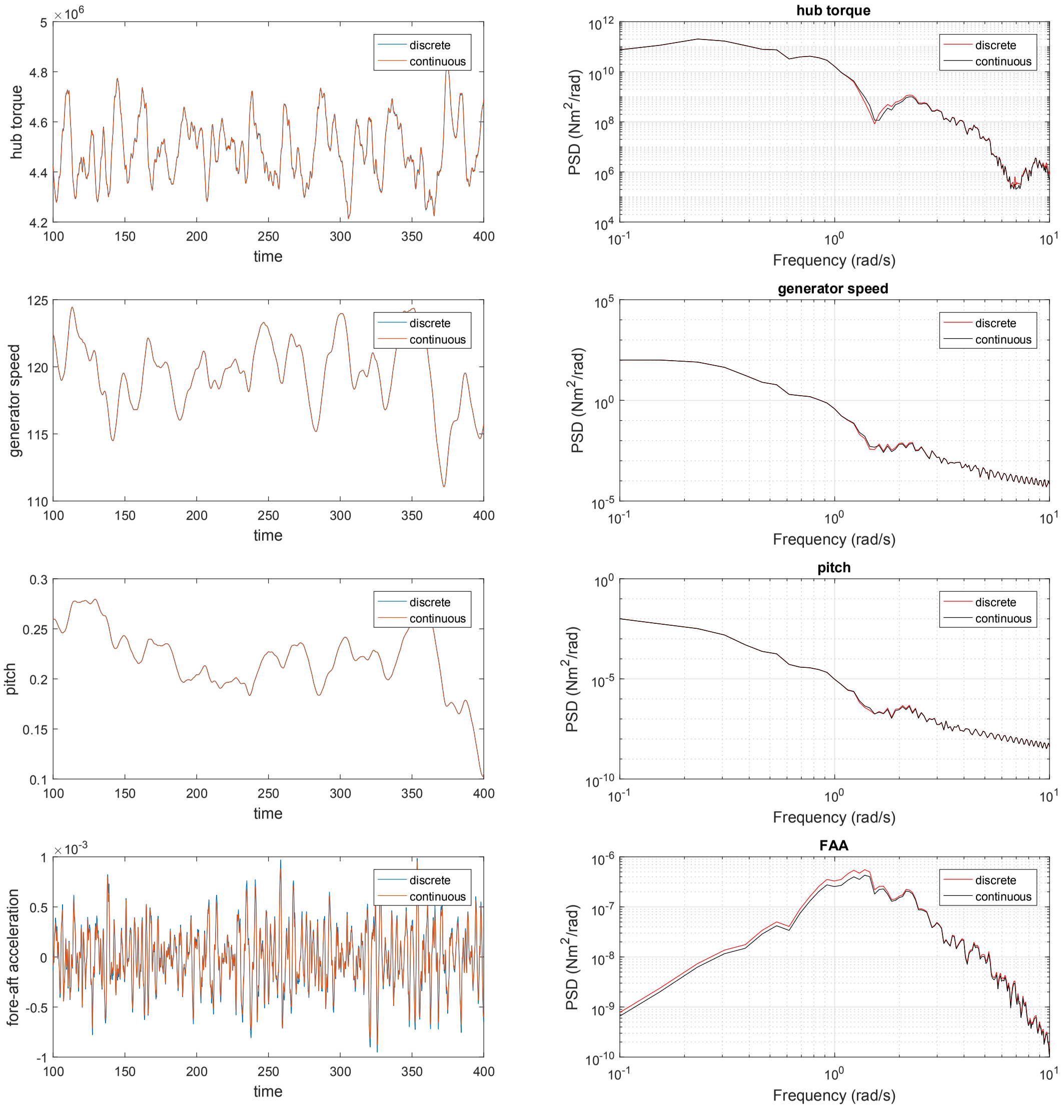


Figure 3: Continuous vs discrete measurements in time and frequency domain.

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 14th British Wind Energy Association (BWEA) Conference, Nottingham*
[9] Leithead W and Connor B 2000 *International Journal of Control* **73** 1189 – 1212

StrathTurb Model Updates

Lindsey Amos

Wind and Marine Energy CDT, University of Strathclyde

E-mail: `lindsey.amos@strath.ac.uk`

June 2019

Contents

1 Introduction **2**

2 Updated Baseline Rotor Model **3**

2.1 Updated Lagrangian 3

2.2 Rotor Dynamics Equations of Motion 4

3 Updated Single Blade Model **6**

3.1 Updated Lagrangian 6

3.2 Individual Blade Model Equations of Motion 7

4 Below Rated Results **8**

5 Above Rated Results **11**

Appendix A Parameters **14**

1. Introduction

This document reports on updates to wind turbine model equations of motion and parameters, made to ensure the model reflects the behaviour of larger, flexible turbines.

Changes have been made to the baseline rotor model and single blade model. From the new Lagrangian's the equations of motion for the rotor dynamics and single blade dynamics are found. With these equations implemented, and the appropriate modifications made to the wind turbine parameter set, the model is then tested. The results of comparison to both the previous wind turbine model and DNV-GL Bladed outputs are given.

2. Updated Baseline Rotor Model

2.1. Updated Lagrangian

$$
\begin{aligned}
\mathcal{L}_R = \frac{1}{2}J\theta_R'^2 + \frac{1}{2}J\phi_R'^2 + J_c\phi_R'\phi_T' + \frac{1}{2}J_T\phi_T'^2 \\
-\frac{1}{2}k_E[(\theta_R - \theta_H)\cos\beta - (\phi_R - K_a\phi_T)\sin\beta]^2 \\
-\frac{1}{2}k_F[(\theta_R - \theta_H)\sin\beta + (\phi_R - K_a\phi_T)\cos\beta]^2 \\
-\frac{1}{2}k_T\phi_T^2 + D_T\phi_T \\
+D_E[(\theta_R - \theta_H)\cos\beta - (\phi_R - K_a\phi_T)\sin\beta] \\
+D_F[(\theta_R - \theta_H)\sin\beta + (\phi_R - K_a\phi_T)\cos\beta] \\
+M_{A\theta_R}\theta_R + M_{A\phi_R}\phi_R \\
-\bar{M}_{A\theta_R}K_a\phi_T + F_t h\phi_T \\
-\frac{1}{2}J\Omega^2(\phi_R - K_a\phi_T)^2 + M_r g R_c\sin(K_a\phi_T)\sin(\phi_R - K_a\phi_T) \\
-M_T g\bar{h}\cos\phi_T cos\theta_T
\end{aligned}
\tag{1}
$$

Where

- $J_c = M_R h R_c - J K_a$ is the tower/rotor cross coupling inertia;
- $J_T = J_{Tower/nacelle} + [\frac{3}{2}K_a^2 J + M_R h(h - 2K_a R_c)]$ is the total inertia of tower, nacelle and rotor;
- $D_T = -B_T\phi_T$ is the tower fore-aft damping force;
- $D_E = -\gamma_E\theta_E' = -\gamma_E(\theta_R'\cos\beta - \phi_R'\sin\beta)$ is the edgewise material damping of the blade;
- $D_F = -\gamma_F\phi_F' = -\gamma_F(\theta_R'\sin\beta + \phi_R'\cos\beta)$ is the flapwise material damping of the blade;
- $k_E = J\omega_{es}^2$ and $k_F = J\omega_{fs}^2$ are the edge-wise and flap-wise stiffnesses of the blade;
- $K_T = J_T\omega_T^2$ is the stiffness of the tower required for the frequency to be that of the fore-aft structural mode of the tower;
- M_T is the total mass of the tower, nacelle and rotor [kg];
- $\bar{h}$ is the height of centre of mass of the tower, nacelle and rotor;
- $\Omega = \theta_R'$ is the nominal angular velocity of the rotor.

The aerodynamic torques $M_{A\theta_R}$ and $M_{A\phi_R}$ are functions of Ω_R, W_R and β, whereas $\bar{M}_{A\phi_R}$ is a function of Ω_R, $(W_R + \frac{1}{2}R_c K_a\phi_T')$ and β.

2.2. Rotor Dynamics Equations of Motion

The equations of motion are derived from the Lagrangian of the system, which combines the contributions to motion that come from kinetic energy, potential energy and virtual work. The rotor dynamics are defined by the following equations:

$$\begin{aligned}\theta_R'' = -\omega_{es}^2 \cos\beta[(\theta_R-\theta_H)\cos\beta-(\phi_R-K_a\phi_T)\sin\beta] \\ -\omega_{fs}^2 \sin\beta[(\theta_R-\theta_H)\sin\beta+(\phi_R-K_a\phi_T)\cos\beta] \\ -\frac{\gamma_E}{J}[\theta_R'\cos\beta-\phi_R'\sin\beta]\cos\beta \\ -\frac{\gamma_F}{J}[\theta_R'\sin\beta+\phi_R'\cos\beta]\sin\beta \\ +\frac{M_{A\theta_R}}{J}\end{aligned} \tag{2}$$

$$\begin{aligned}\frac{JJ_T-J_c^2}{JJ_T+K_aJJ_c}\phi_R'' = \omega_{es}^2 \sin\beta[(\theta_R-\theta_H)\cos\beta-(\phi_R-K_a\phi_T)\sin\beta] \\ -\omega_{fs}^2 \cos\beta[(\theta_R-\theta_H)\sin\beta+(\phi_R-K_a\phi_T)\cos\beta] \\ +\frac{M_rgR_c}{J}\sin(K_a\phi_T)\cos(\phi_R-K_a\phi_T) \\ -\frac{J_cK_a}{J(J_T+K_aJ_c)}M_rgR_c\cos(K_a\phi_T)\sin(\phi_R-K_a\phi_T) \\ -\frac{J_c}{J(J_T+K_aJ_c)}M_tg\bar{h}\sin\phi_T\cos\theta_T \\ +\frac{\gamma_E}{J}\sin\beta[\theta_R'\cos\beta-\phi_R'\sin\beta] \\ +\frac{\gamma_F}{J}\cos\beta[\theta_R'\sin\beta+\phi_R'\cos\beta] \\ +\frac{J_c}{J(J_T+K_aJ_c)}(K_T\phi_T+B_T\phi_T') \\ -(\omega_f^2-\omega_{fs}^2)(\frac{\theta_R'^2}{\Omega_R^2})(\phi_R-K_a\phi_T) \\ +\frac{1}{J(J_T+K_aJ_c)}[J_TM_{A\phi_R}+J_cK_a\bar{M}_{A\phi_R}-J_chF_T]\end{aligned} \tag{3}$$

$$\begin{aligned}
\frac{JJ_T - J_c^2}{JJ_c + K_a J^2}\phi_T'' = &-\omega_{es}^2 \sin\beta[(\theta_R - \theta_H)\cos\beta - (\phi_R - K_a\phi_T)\sin\beta] \\
&+\omega_{fs}^2 \cos\beta[(\theta_R - \theta_H)\sin\beta + (\phi_R - K_a\phi_T)\cos\beta] \\
&-\frac{\gamma_E}{J}\sin\beta[\theta_R'\cos\beta - \phi_R'\sin\beta] \\
&+\frac{\gamma_F}{J}\cos\beta[\theta_R'\sin\beta + \phi_R'\cos\beta] \\
&-\frac{M_r g R_c}{J}\sin(K_a\phi_T)\cos(\phi_R - K_a\phi_T) \\
&+\frac{M_r g R_c K_a}{(J_c + K_a J)}\cos(K_a\phi_T)\sin(\phi_R - K_a\phi_T) \\
&+\frac{M_T g\bar{h}}{(J_c + K_a J)}\sin\phi_T\cos\theta_T \\
&+(\omega_f^2 - \omega_{fs}^2)(\frac{\theta_R'^2}{\Omega_R^2})(\phi_R - K_a\phi_T) \\
&-\frac{1}{(J_c + K_a J)}(K_T\phi_T + B_T\phi_T' - hF_t) - \frac{J_c}{J(J_c + K_a J)}M_{A\phi_R} \\
&-\frac{K_a}{(J_c + K_a J)}\bar{M}_{A\phi_R}
\end{aligned} \tag{4}$$

And the hub torque, defined as $T_H = M_{A\theta_R} - J\theta_R''$, is:

$$\begin{aligned}
T_H = &J\omega_{es}^2 \cos\beta[(\theta_R - \theta_H)\cos\beta - (\phi_R - K_a\phi_T)\sin\beta] \\
&+J\omega_{fs}^2 \sin\beta[(\theta_R - \theta_H)\sin\beta + (\phi_R - K_a\phi_T)\cos\beta] \\
&+\gamma_E[\theta_R'\cos\beta - \phi_R'\sin\beta]\cos\beta \\
&+\gamma_F[\theta_R'\sin\beta + \phi_R'\cos\beta]\sin\beta
\end{aligned} \tag{5}$$

3. Updated Single Blade Model

3.1. Updated Lagrangian

$$
\begin{aligned}
\mathcal{L}_{bld} = {} & \frac{1}{2}J_b(\theta'_R + K_a\theta'_{Tb} + \phi'\cos\phi_{Rb})^2 + \frac{1}{2}J_b(K_a\phi'_{Tb}\cos\psi + \phi'_{Rb})^2 + J_b\psi' K_a\phi'_{Tb}\sin\phi_{Rb}\sin\psi \\
& +\frac{1}{2}M_b h^2\theta'^2_{Tb} + \frac{1}{2}M_b h^2\phi'^2_{Tb} + M_bR_ch\theta'_{Tb}\cos\psi(\theta'_{Rb} + \psi') + M_bR_ch\phi'_{Rb}\phi'_{Tb} \\
& -M_bg[R_c(\cos(K_a\phi_{Tb})\cos(K_a\theta_{Tb})\cos\psi\cos(\theta_{Rb})\cos(\phi_{Rb}) \\
& - \sin\psi\sin(K_a\theta_{Tb}) - \sin\psi\sin\theta_{Rb} - \sin(K_a\theta_{Tb})\cos\psi\sin\theta_{Rb} \\
& - \sin(K_a\phi_{Tb})\sin\phi_{Rb}) + \cos\phi_{Tb}\cos\theta_{Tb}h] \\
& -\frac{1}{2}k_E(\theta_{Rb}\cos\beta - \phi_{Rb}\sin\beta)^2 - \frac{1}{2}k_F(\theta_{Rb}\sin\beta + \phi_{Rb}\cos\beta)^2 \\
& +D_{bE}(\theta_{Rb}\cos\beta - \phi_{Rb}\sin\beta) + D_{bF}(\theta_{Rb}\sin\beta + \phi_{Rb}\cos\beta) \\
& +M_{A\theta_{Rb}}(K_a\theta_{Tb} + \theta_{Rb} + \psi) + M_{A\phi_{Rb}}(K_a\phi_{Tb}\cos(\theta_{Rb} + \psi) + \phi_{Rb}) + F_{bT}h\phi_{Tb}
\end{aligned}
\tag{6}
$$

Where blade material damping includes the terms $D_{bE} = -\gamma_E(\theta'_{Rb}\cos\beta - \phi'_{Rb}\sin\beta)$ and $D_{bF} = -\gamma_F(\theta'_{Rb}\sin\beta + \phi'_{Rb}\cos\beta)$.

The approximate relationships between the rotor and single blade motions are:

(i) Tower motion:

- $\theta_{Tb} = \theta_T$; $\theta'_{Tb} = \theta'_T$; $\theta''_{Tb} = \theta''_T$
- $\phi_{Tb} = \phi_T + \epsilon$; $\phi'_{Tb} \approx \phi'_T$; $\phi''_{Tb} \approx \phi''_T$

(ii) Single blade and hub motion:

- $\theta_R \approx \theta_{Tb} + \theta_{Rb} + \psi$; $\phi_R \approx \phi_{Tb} + \phi_{Rb}$
- $\theta'_R = \theta'_{Tb} + \theta'_{Rb}\cos(\phi_{Tb}) + \psi'\cos(\phi_{Tb}) + \phi'_{Rb}\sin(\phi_{Tb})\sin(\psi) \approx \theta'_{Tb} + \theta'_{Rb} + \psi'\cos(\phi_{Tb}) + \phi'_{Rb}\sin(\phi_{Tb})\sin(\psi)$
- $\theta''_H \approx \psi'' + \theta''_{Tb}$

(iii) Rotor motion

- $\theta_R \approx \theta_{Tb} + \theta_{Rb} + \psi$
- $\theta'_R \approx \theta'_{Tb} + \theta'_{Rb} + \psi'\cos(\phi_{Tb})$; $\theta'_H - \theta'_T = \psi'\cos(\phi_{Tb})$
- $\phi_R \approx \phi_{Tb} + \phi_{Rb}$

Note: ϕ_{Tb} has been corrected for tilt (ϵ).

3.2. Individual Blade Model Equations of Motion

The equations of motion are derived from the Lagrangian of the system, which combines the contributions to motion that come from kinetic energy, potential energy and virtual work. The dynamics of the individual blade are defined by the following equations of motion:

$$\begin{aligned}
\theta''_{Rb} = {} & \phi_{Rb}\sin\beta\cos\beta[\omega_{es}^2 - \omega_{fs}^2] - \theta_{Rb}[\sin^2\beta\omega_{fs}^2 + \cos^2\beta\omega_{es}^2] \\
& + \phi'_{Rb}\sin\beta\cos\beta[\gamma_E - \gamma_F] - \theta'_{Rb}[\gamma_F\sin^2\beta + \gamma_E\cos^2\beta] \\
& - \frac{gM_bR_c}{J_b}[\sin\psi\cos\theta_{Rb} + \cos\psi(\cos(K_a\phi_{Tb})\cos(K_a\theta_{Tb})\sin\theta_{Rb}\cos\phi_{Rb} + \sin(K_a\theta_{Tb})\cos\theta_{Rb})] \\
& - \psi''\cos\phi_{Rb} - \theta''_{Tb}(K_a + \cos\psi\frac{M_bR_ch}{J_b}) \\
& \frac{M_{A\theta_{Rb}}}{J_b} - \frac{M_{A\phi_{Rb}}}{J_b}(K_a\phi_{Tb}\sin(\psi + \theta_{Rb}))
\end{aligned} \tag{7}$$

$$\begin{aligned}
\phi''_{Rb} = {} & \theta_{Rb}\sin\beta\cos\beta[\omega_{es}^2 - \omega_{fs}^2] - \phi_{Rb}[\sin^2\beta\omega_{es}^2 + \cos^2\beta\omega_{fs}^2] \\
& + \theta'_{Rb}\sin\beta\cos\beta[\gamma_E - \gamma_F] - \phi'_{Rb}[\gamma_E\sin^2\beta + \gamma_F\cos^2\beta] \\
& + \frac{gM_bR_c}{J_b}(\cos\psi\sin\phi_{Rb} + \sin(K_a\phi_{Tb}) \\
& - \phi''_{Tb}(K_a + \frac{M_bR_ch}{J_b}) \\
& - \psi'\sin\phi_{Rb}(\theta'_{Rb} + K_a\theta'_{Tb}) \\
& + \psi'K_a\phi'_{Tb}\sin\psi\cos\phi_{Rb} - \psi'^2\sin\phi_{Rb} \\
& + \frac{M_{A\phi_{Rb}}}{J_b}
\end{aligned} \tag{8}$$

The blade azimuthal angle, ψ, is calculated from it's derivative (the same as the ψ defined in Rotor Dynamics), where:

$$\psi'' = \theta''_H - \theta''_{Tb} \tag{9}$$

$$\psi' = \frac{\theta'_H - \theta'_{Tb}}{\cos\phi_{Tb}} \tag{10}$$

4. Below Rated Results

Bladed: Mean wind speed = 8.00 m/s; Turbulence intensity = 9.7%

Simulink: Mean wind speed = 8.15 m/s; Turbulence intensity = 9.6%

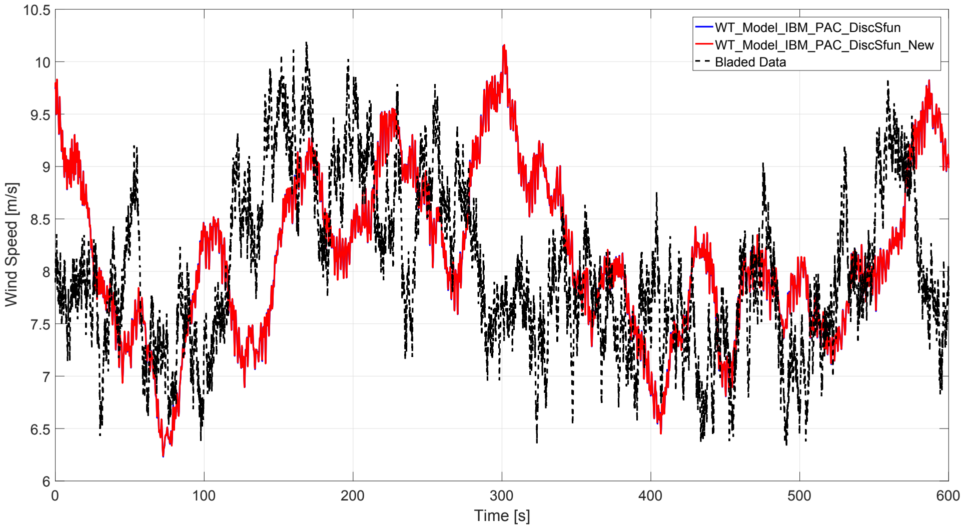


Figure 1. Wind Speed Time Series

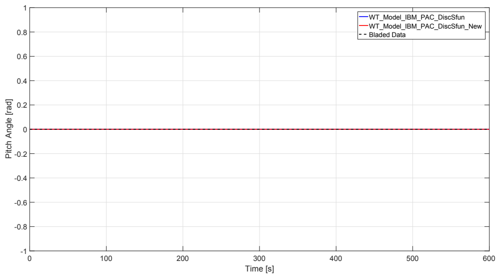


Figure 2. Pitch Angle Time Series

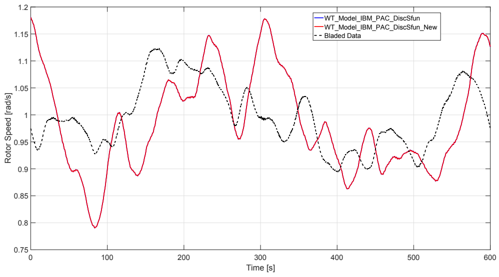


Figure 3. Rotor Speed Time Series

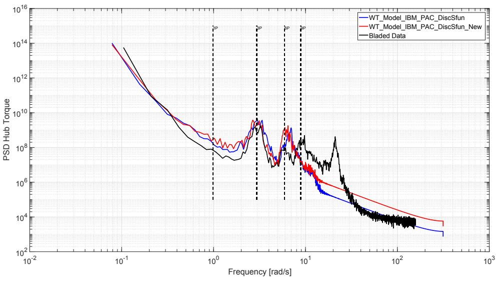


Figure 4. Hub Torque Spectra

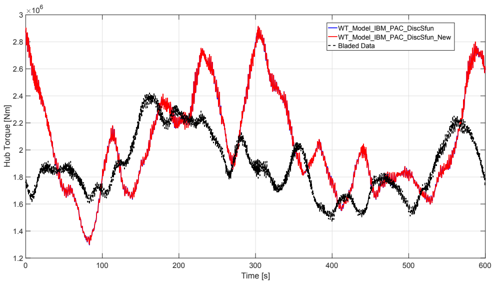


Figure 5. Hub Torque Time Series

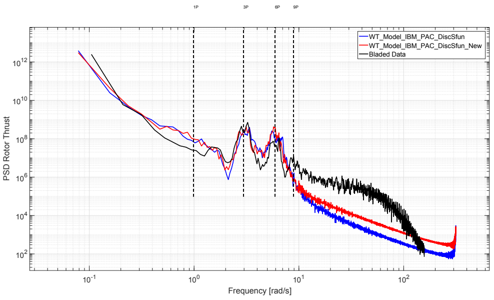


Figure 6. Rotor Thrust Spectra

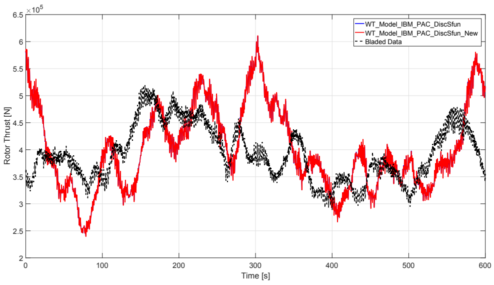


Figure 7. Rotor Thrust Time Series

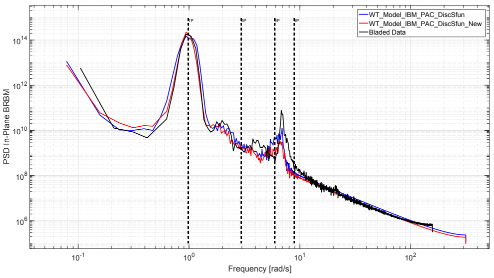


Figure 8. In Plane BRBM Spectra

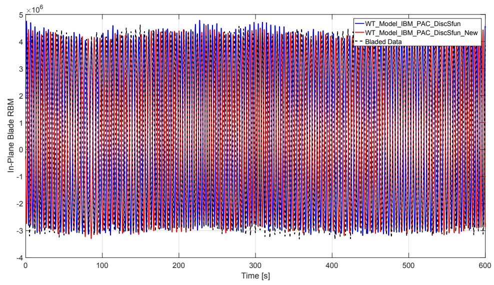


Figure 9. In Plane BRBM Time Series

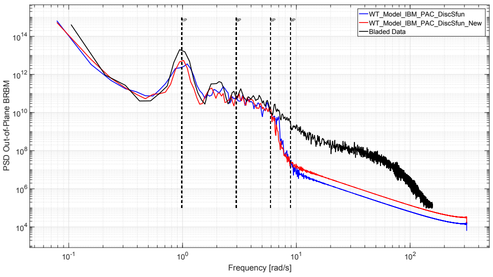


Figure 10. Out-of Plane BRBM Spectra

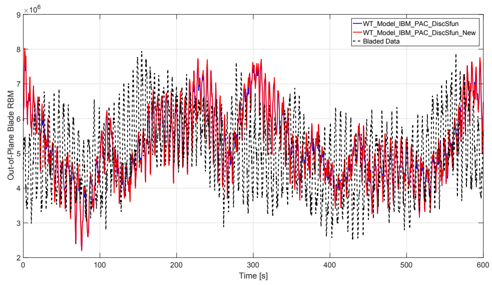


Figure 11. Out-of Plane BRBM Time Series

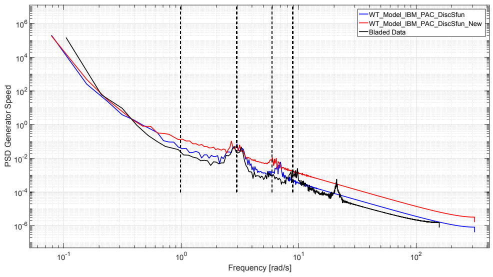


Figure 12. Generator Speed Spectra

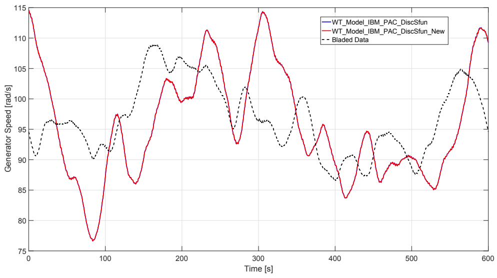


Figure 13. Generator Speed Time Series

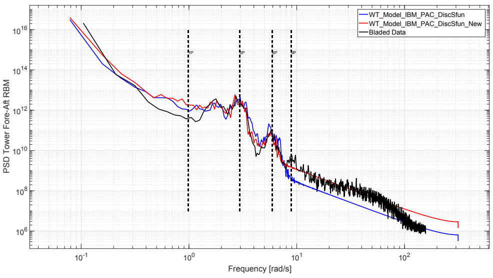


Figure 14. Tower Fore-Aft RBM Spectra

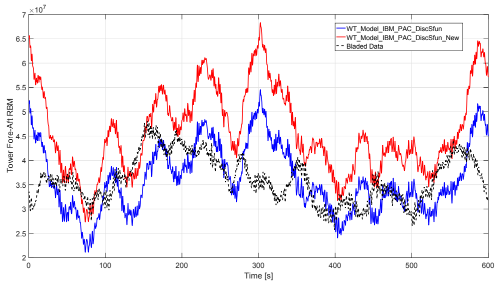


Figure 15. Tower Fore-Aft RBM Time Series

5. Above Rated Results

Bladed: Mean wind speed = 15.00 m/s; Turbulence intensity = 9.7%

Simulink: Mean wind speed = 15.02 m/s; Turbulence intensity = 10.2%

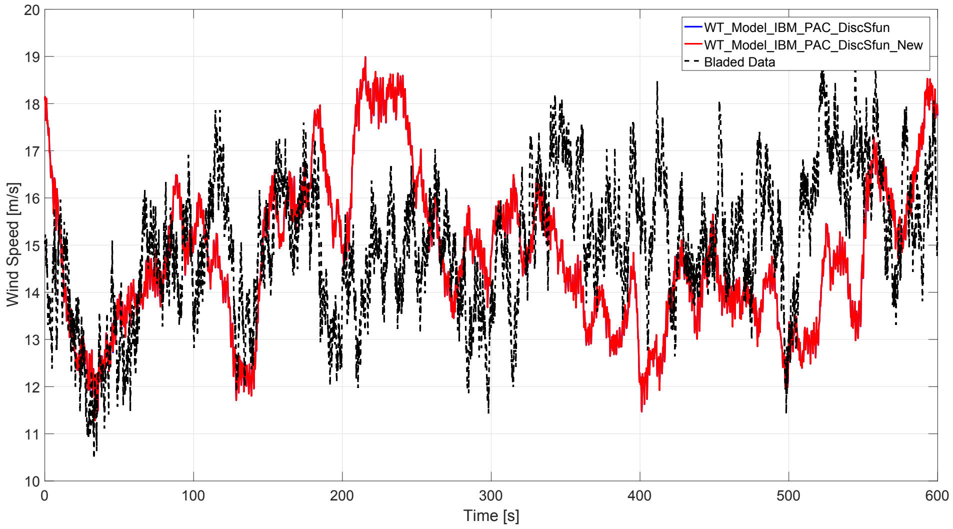


Figure 16. Wind Speed

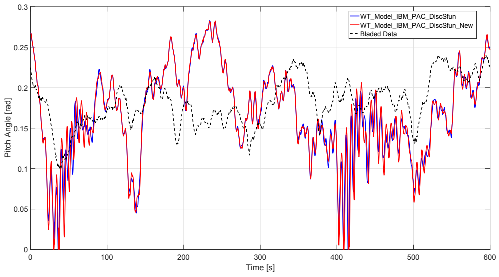


Figure 17. Pitch Angle Time Series

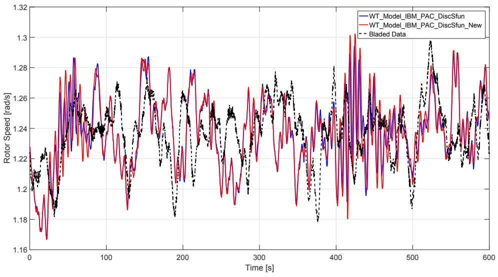


Figure 18. Rotor Speed Time Series

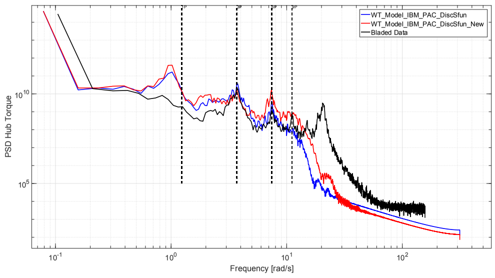


Figure 19. Hub Torque Spectra

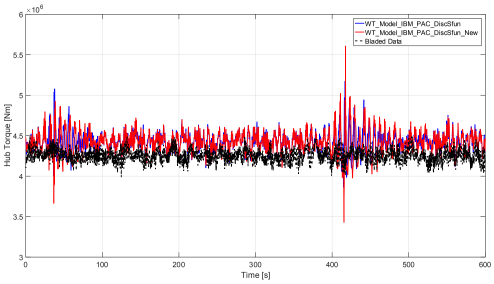


Figure 20. Hub Torque Time Series

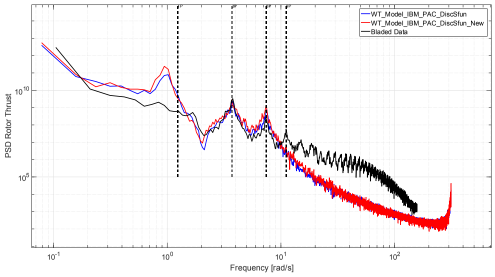


Figure 21. Rotor Thrust Spectra

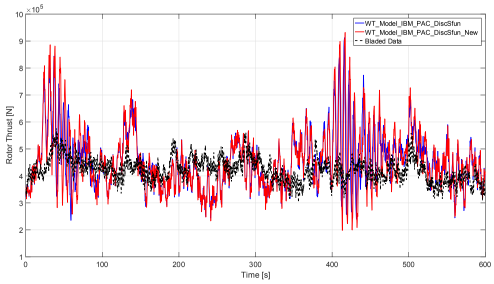


Figure 22. Rotor Thrust Time Series

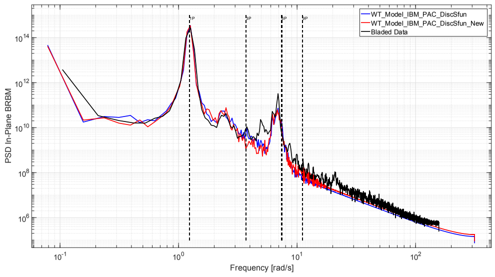


Figure 23. In Plane BRBM Spectra

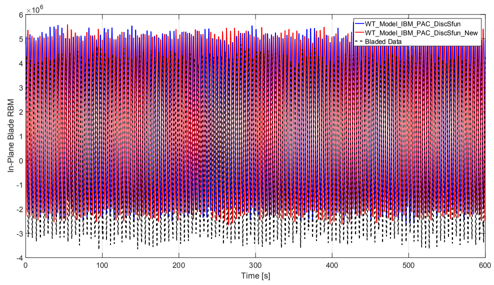


Figure 24. In Plane BRBM Time Series

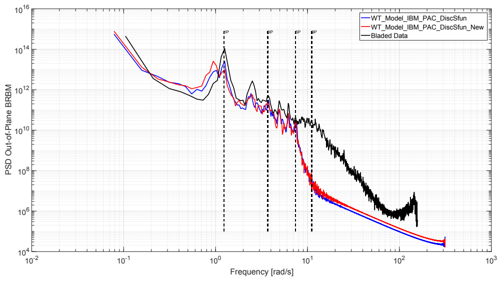


Figure 25. Out-of Plane BRBM Spectra

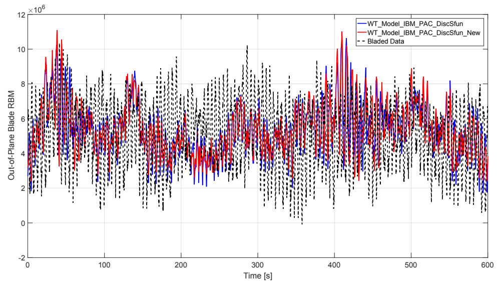


Figure 26. Out-of Plane BRBM Time Series

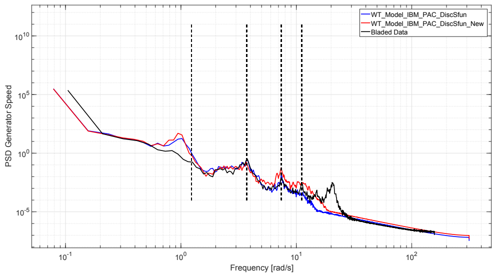


Figure 27. Generator Speed Spectra

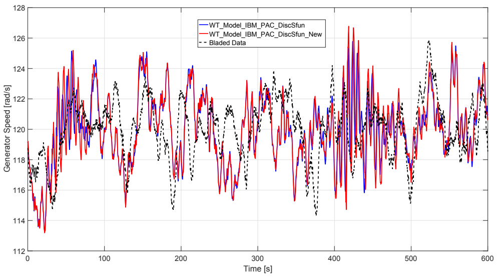


Figure 28. Generator Speed Time Series

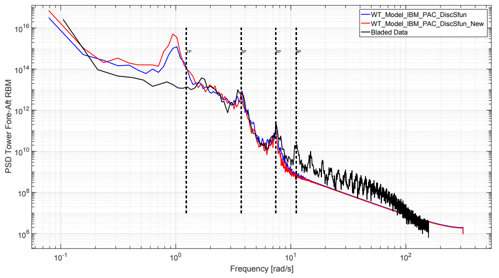


Figure 29. Tower Fore-Aft RBM Spectra

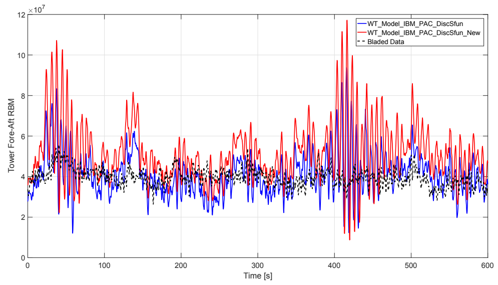


Figure 30. Tower Fore-Aft RBM Time Series