DEPARTMENT OF ELECTRONIC & ELECTRICAL ENGINEERING

University of Strathclyde Engineering

SUPERGEN WIND

EPSRC Engineering and Physical Sciences Research Council

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

REF UK TOP 20 RESEARCH-INTENSIVE UNIVERSITY

THE UK UNIVERSITY OF THE YEAR WINNER

THE UK ENTREPRENEURIAL UNIVERSITY OF THE YEAR WINNER

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.

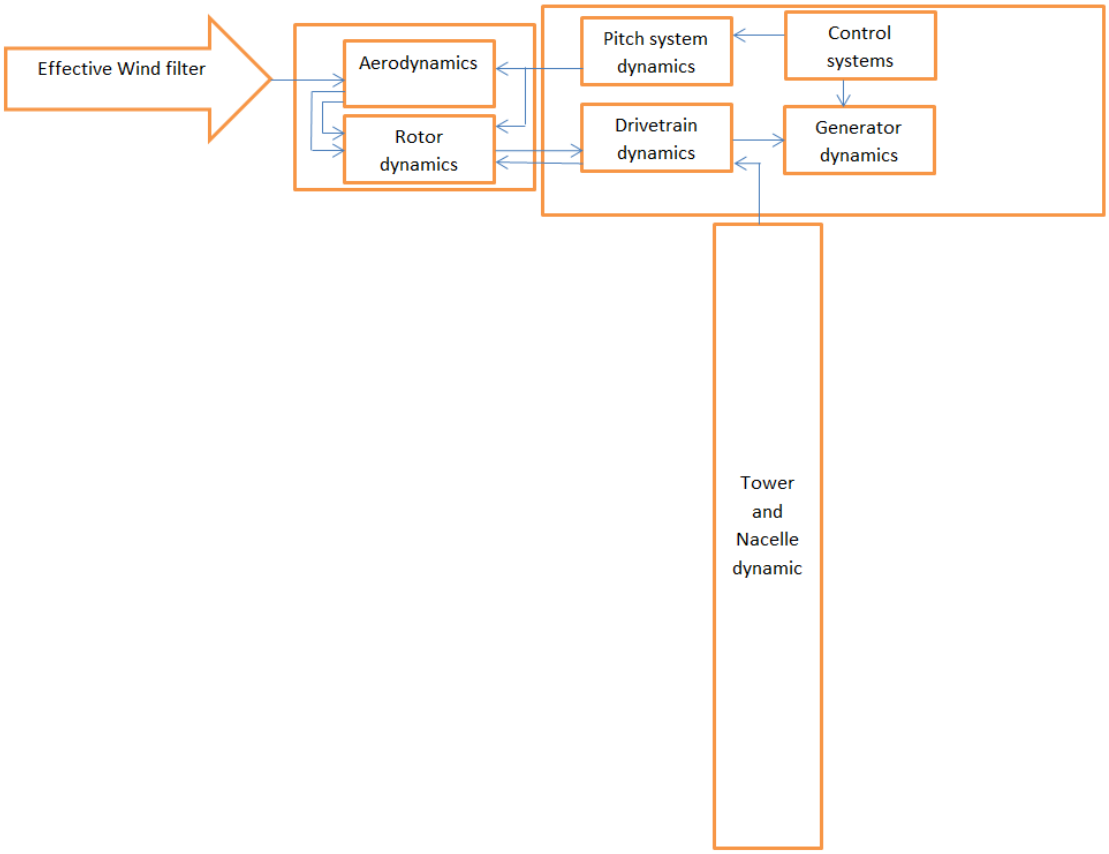

Figure 1: Wind turbine model [1].

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].

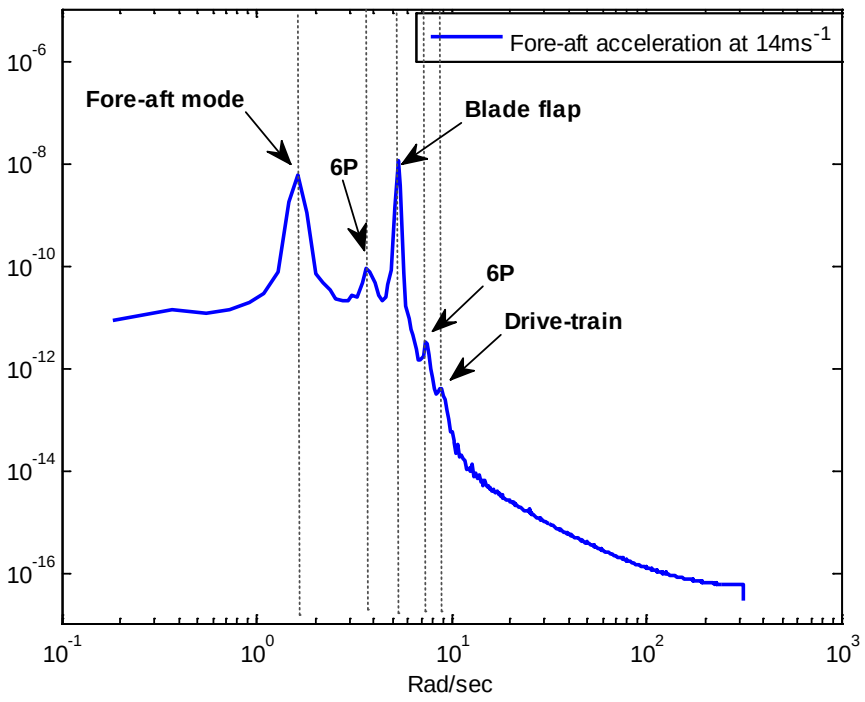

Figure 2: Power spectrum of the tower fore-aft acceleration [1].

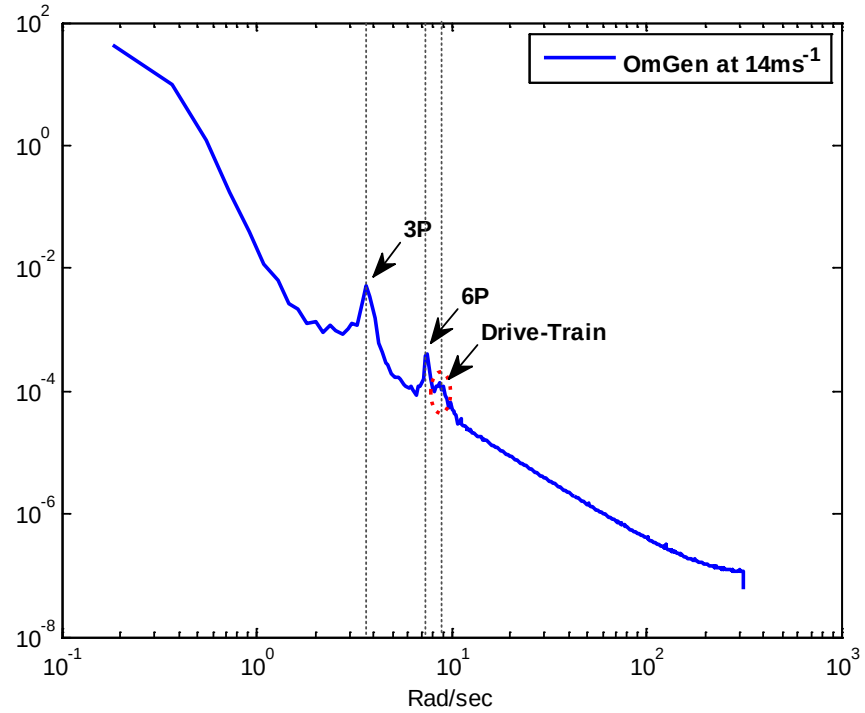

Figure 3: Power spectrum of the tower fore-aft acceleration [1].

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).

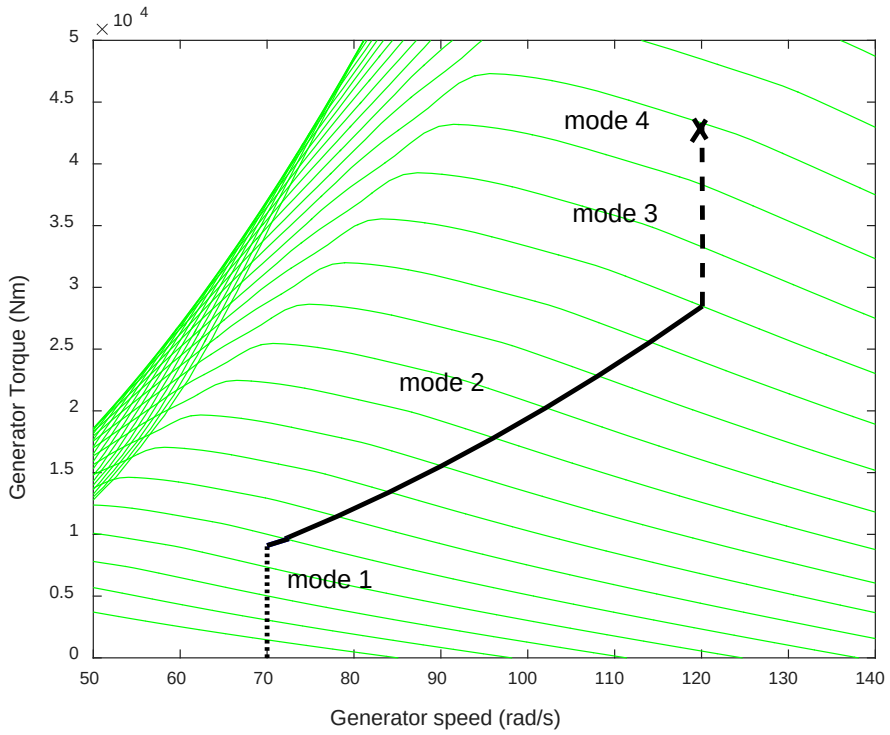

Figure 4: Design operating curve over the full envelope operation on the torque/speed plane.

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].

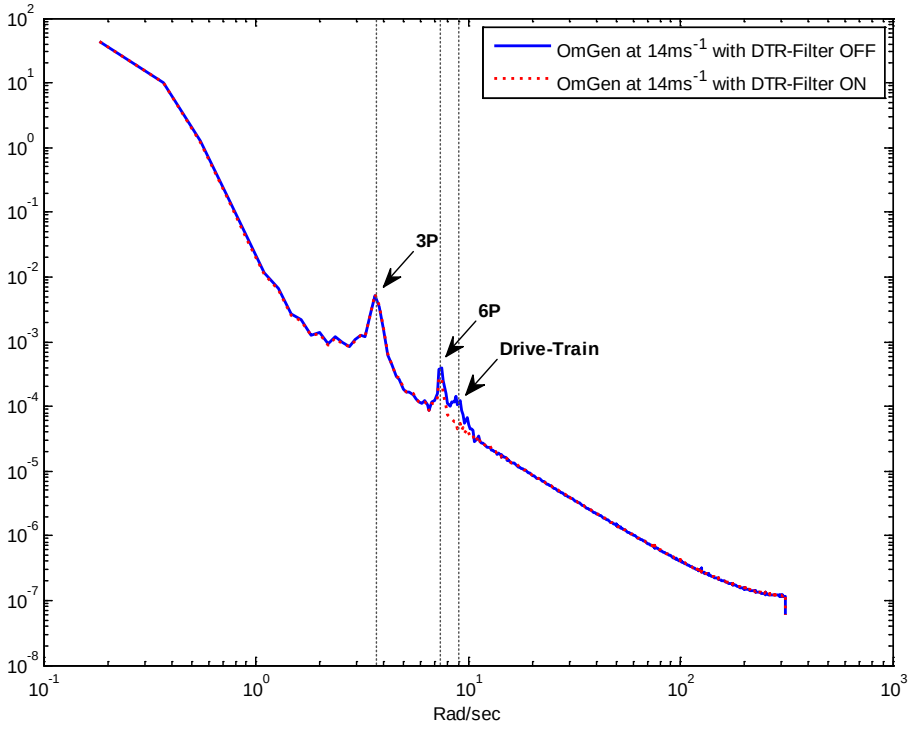

Figure 5: Power Spectral Density of generator speed [1].

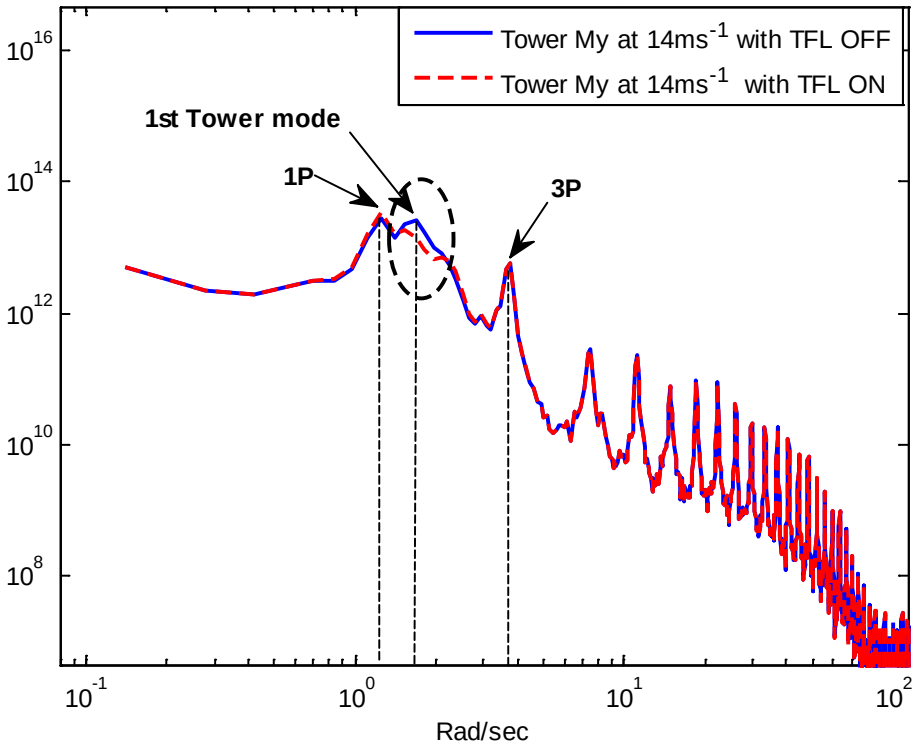

Figure 6: Power Spectral Density of fore-aft tower acceleration [1].

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.

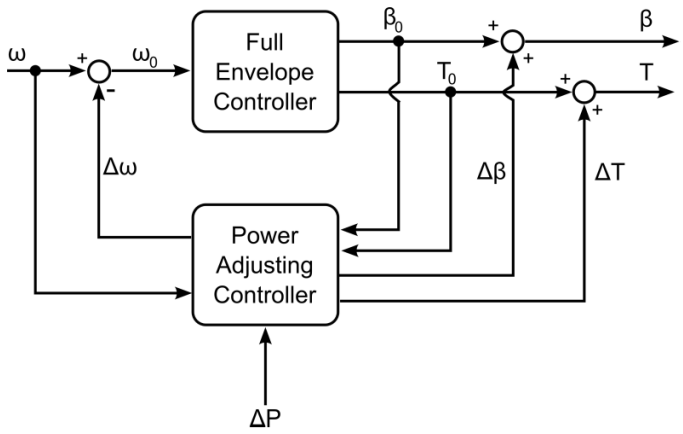

Figure 7: Structure of the PAC [1].

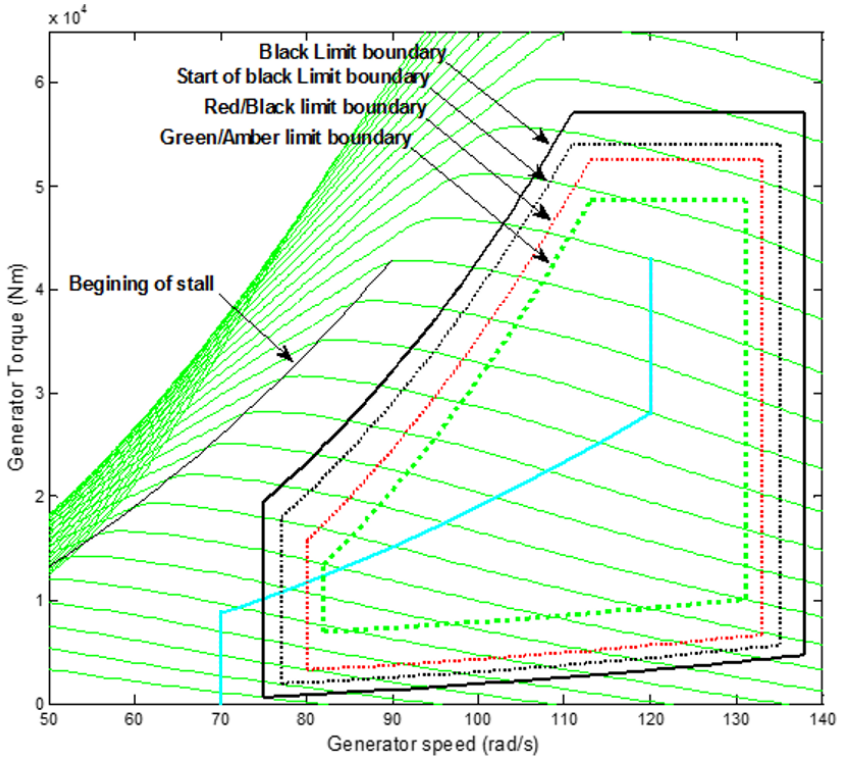

Figure 8: Operating boundaries for the PAC [1].

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".

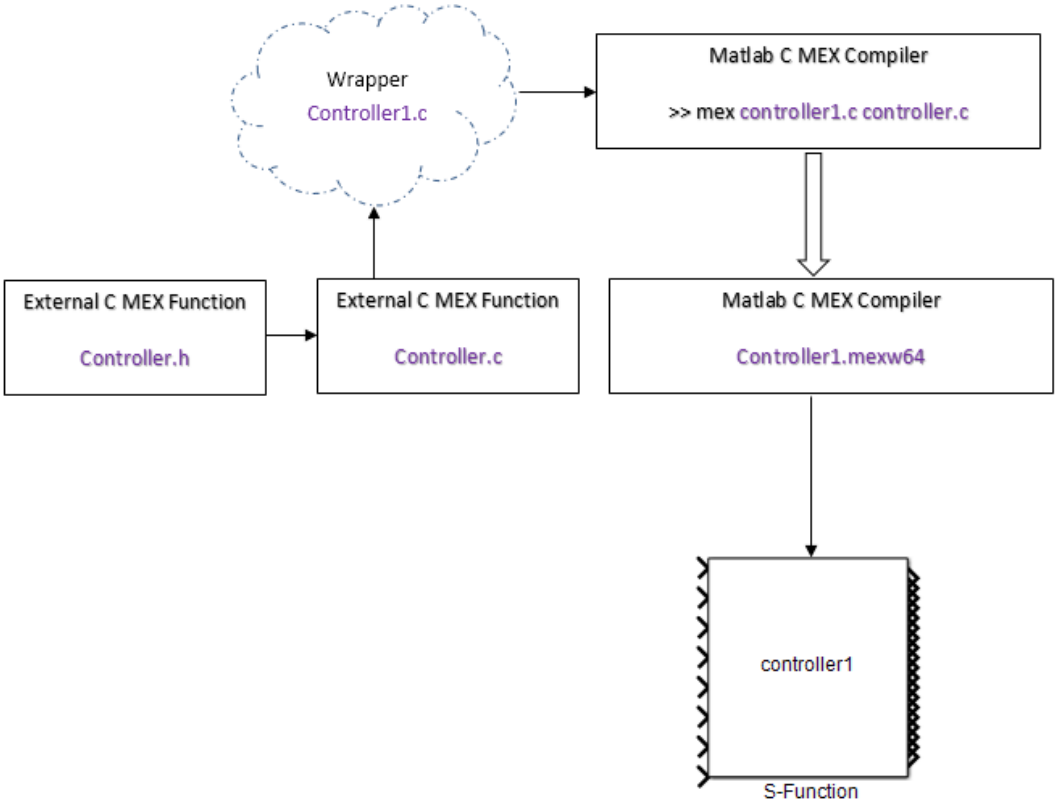

Figure 9: Compiling process of the controller written in C++ [1].

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.

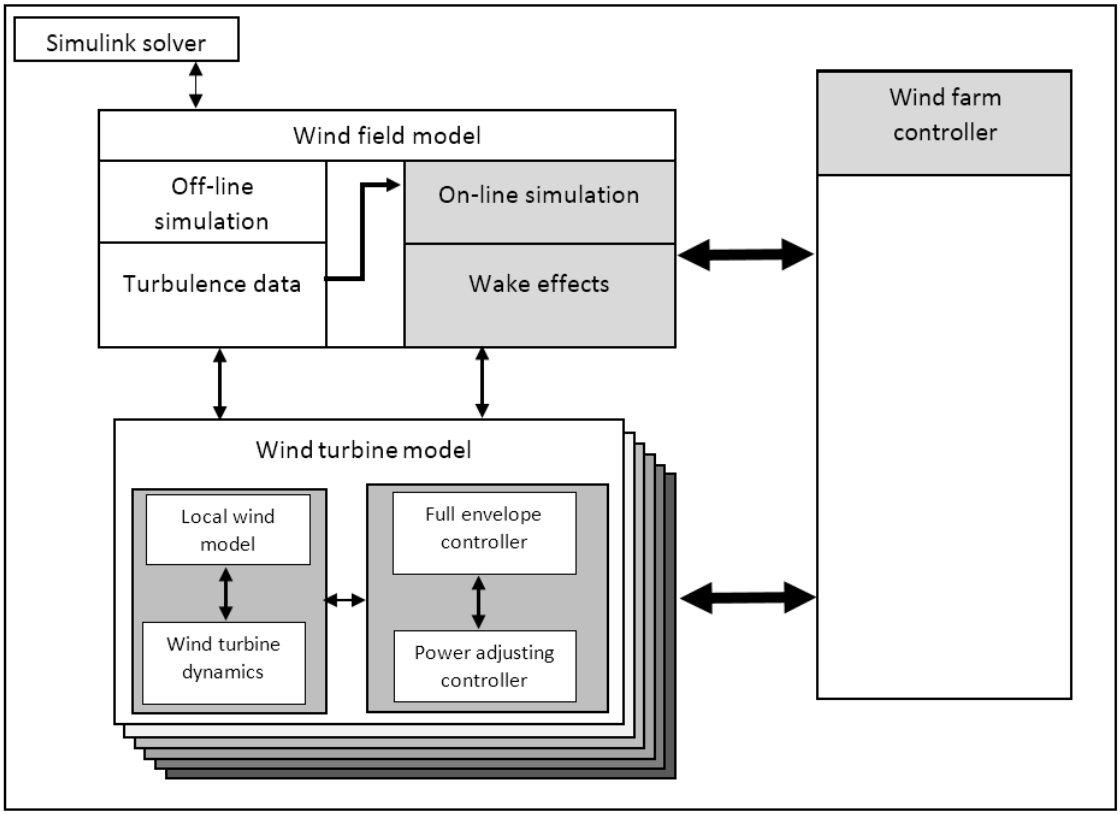

Figure 10: Schematic diagram of the wind farm model [1].

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.

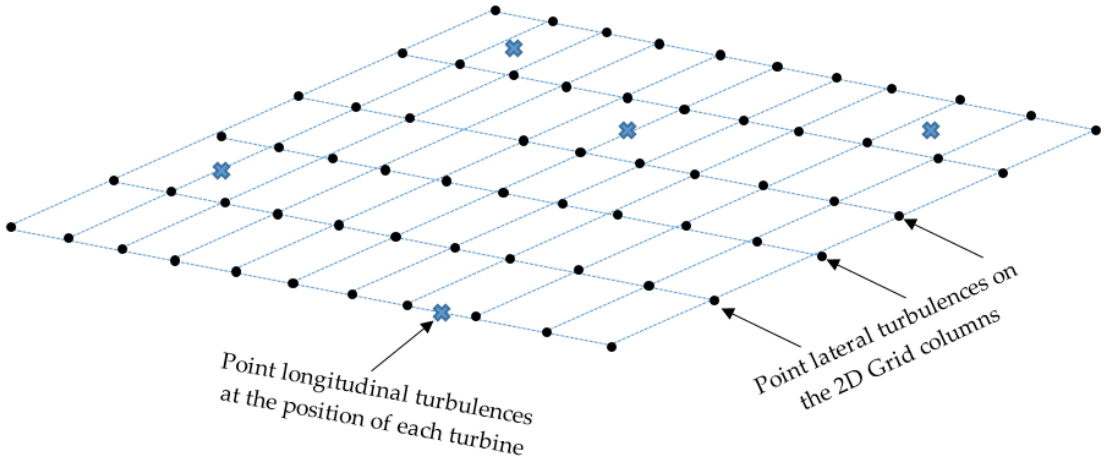

Figure 11: 2D grid layout [1].

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.

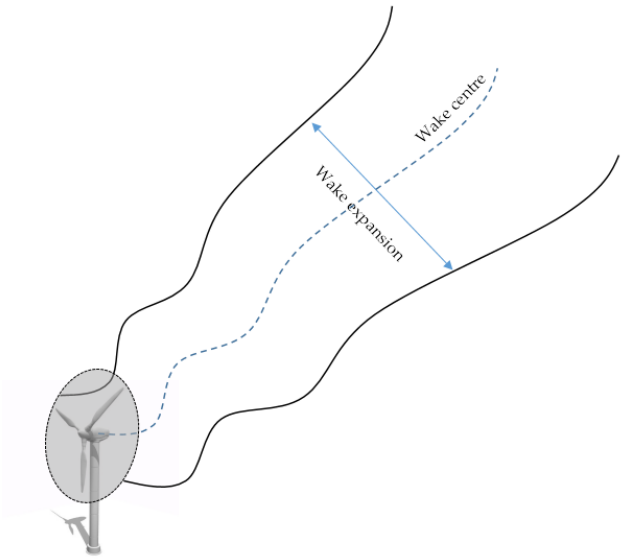

Figure 12: Wake meandering and wake centre [1].

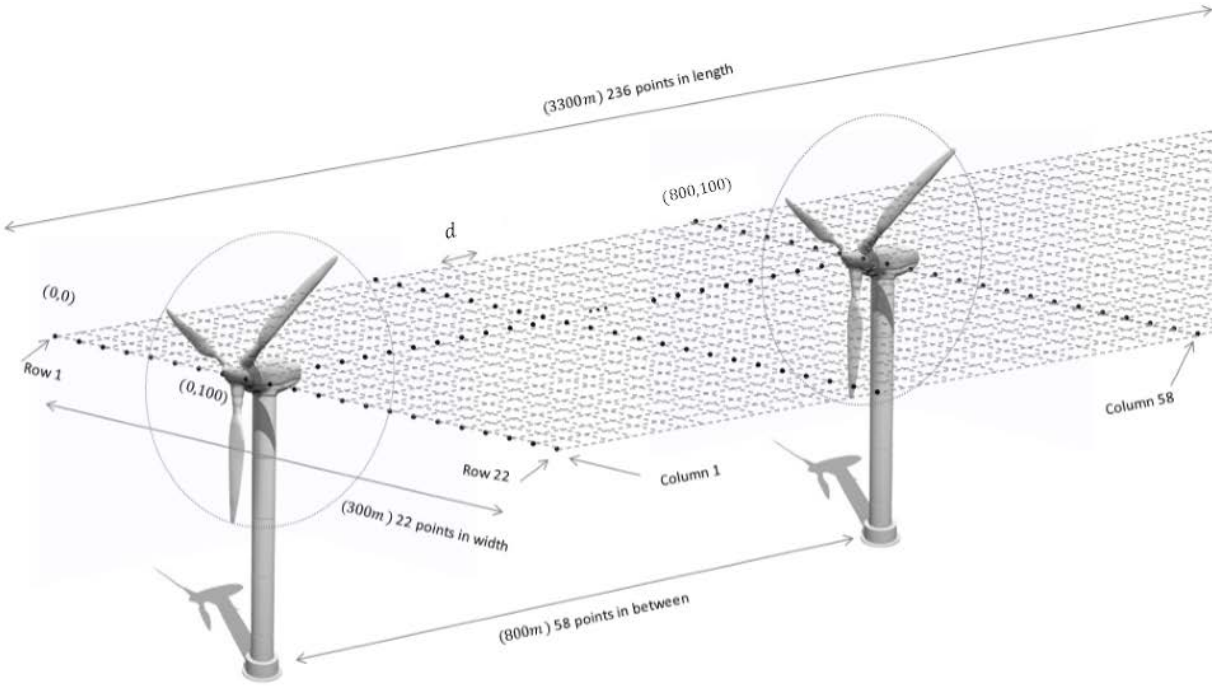

Figure 13: 2D grid layout of the wind farm [1].

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.

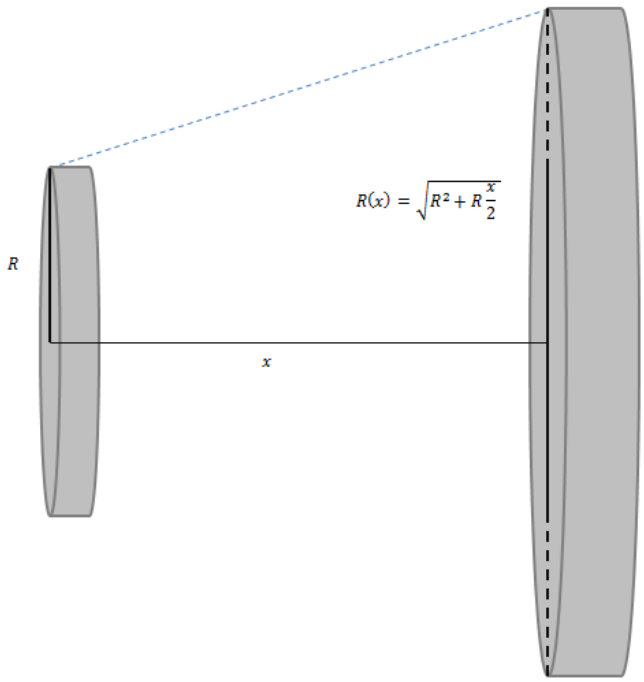

Figure 14: Wake meandering distance [1].

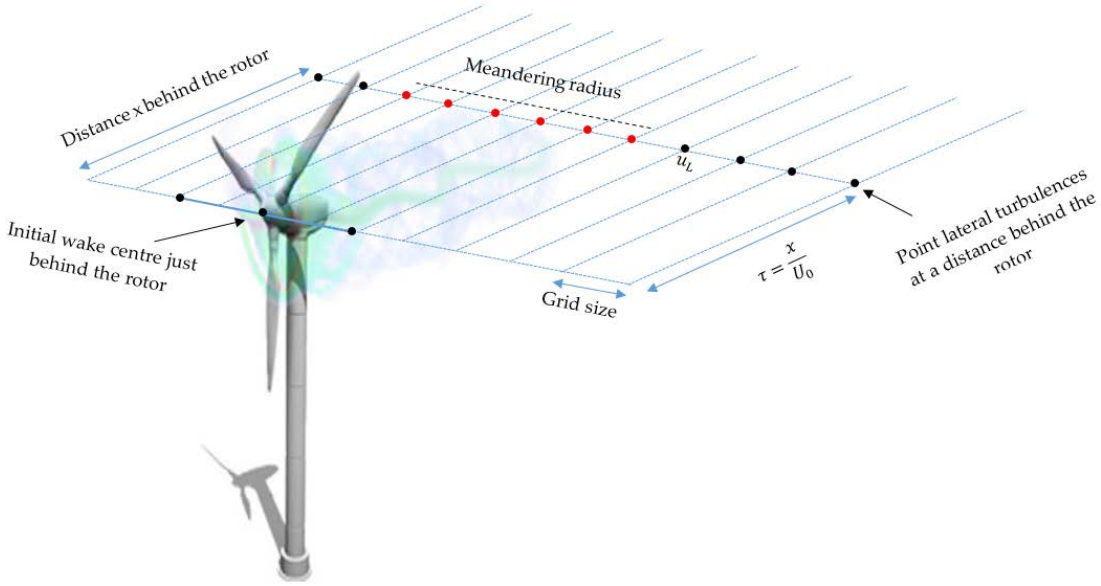

Figure 15: 2D grid layout with grid points inside the meandering radius at distance x [1].

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.

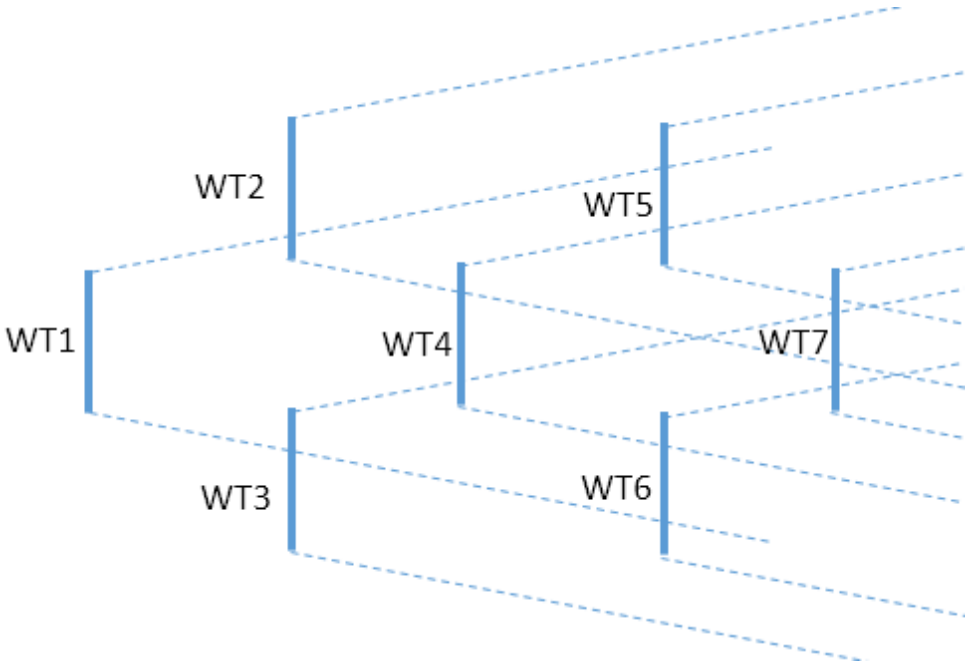

Figure 16: Multiple wake effects [1].

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

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

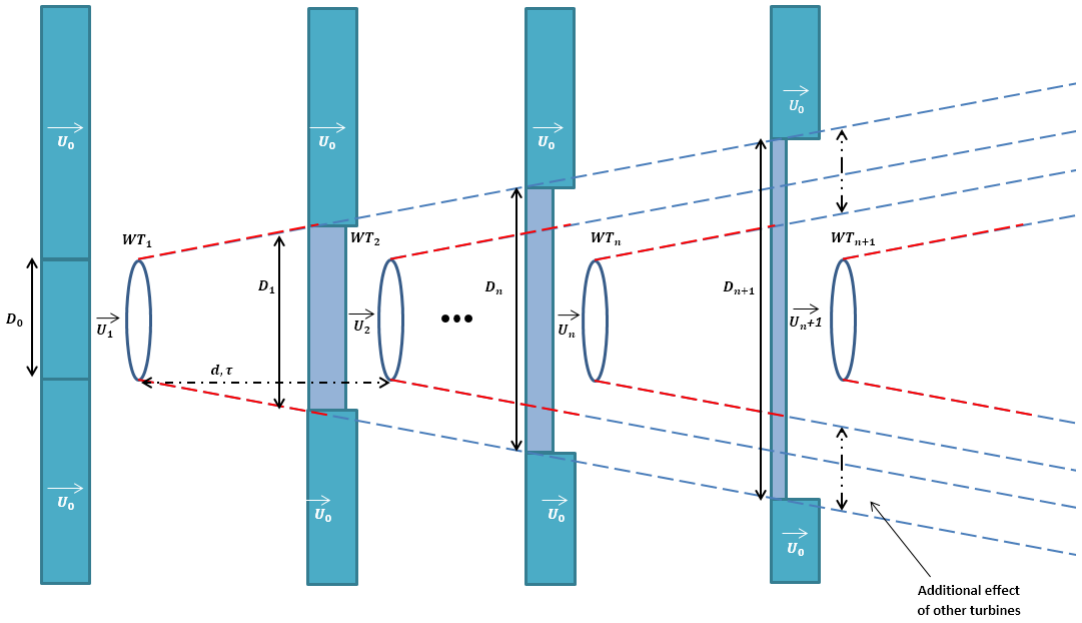

Figure 17: Effect of wake deficit [1].

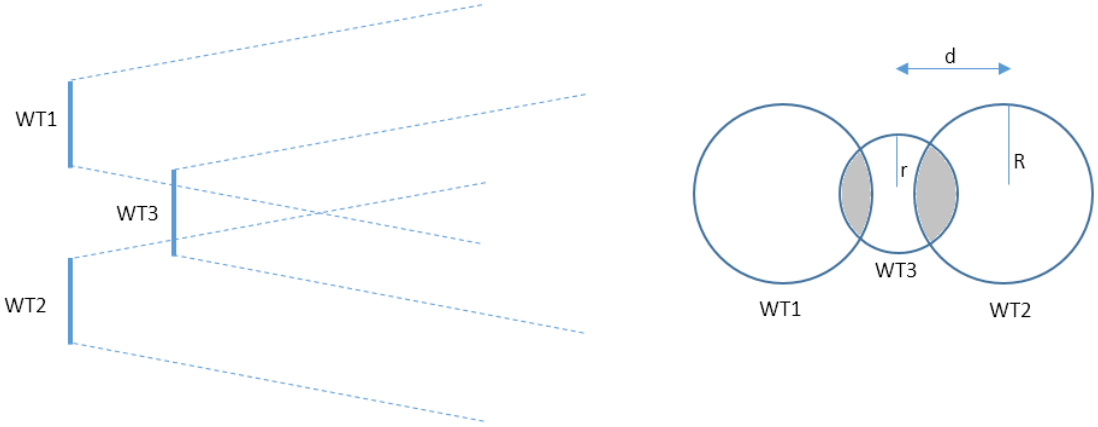

Figure 18: Partial wake shadowing effect [1].

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

$$\mu = \frac{\sinh\big(a(t-t_i)\big)\,x\big(t_f\big) + \sinh\Big(a\big(t_f-t\big)\Big)\,x(t_i)}{\sinh(a(t_f-t_i)} \tag{20}$$

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

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.

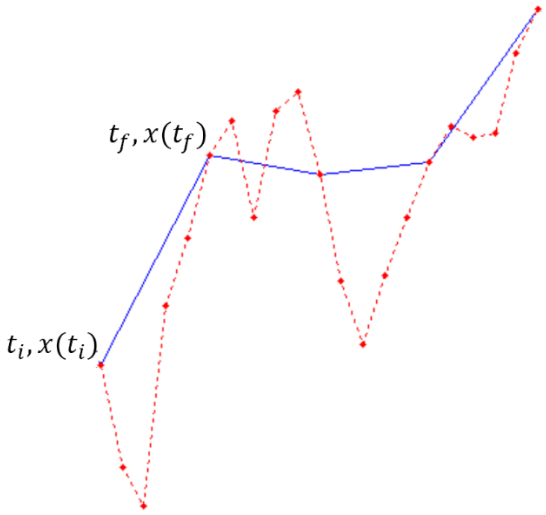

Figure 19: Dryden interpolation [1].

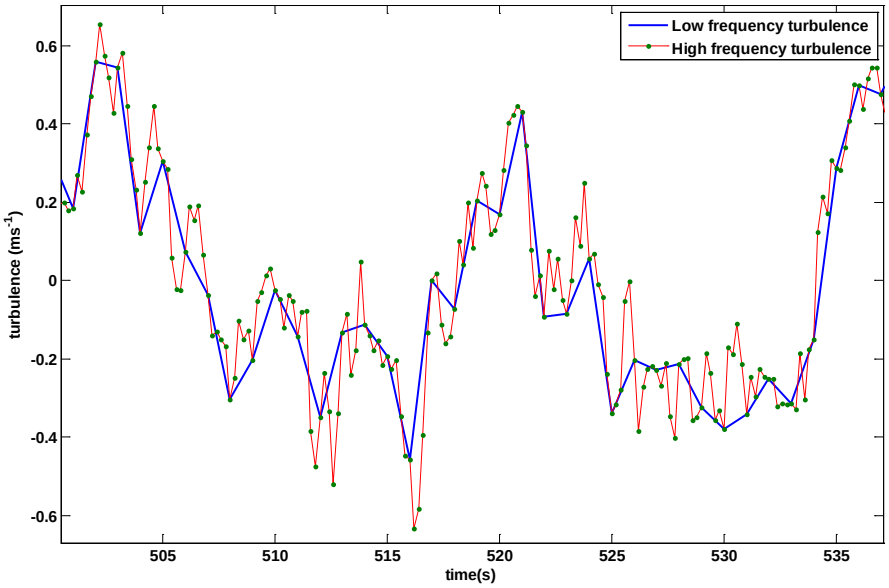

Figure 20: High frequency turbulence [1].

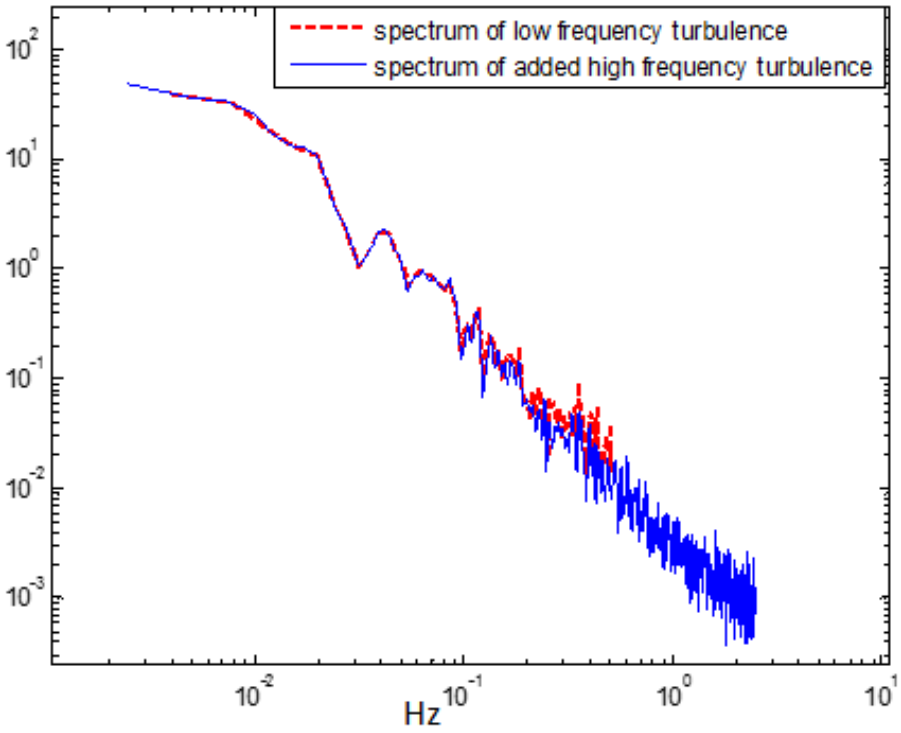

Figure 21: Spectra of the low and high frequency turbulence time series [1].

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.

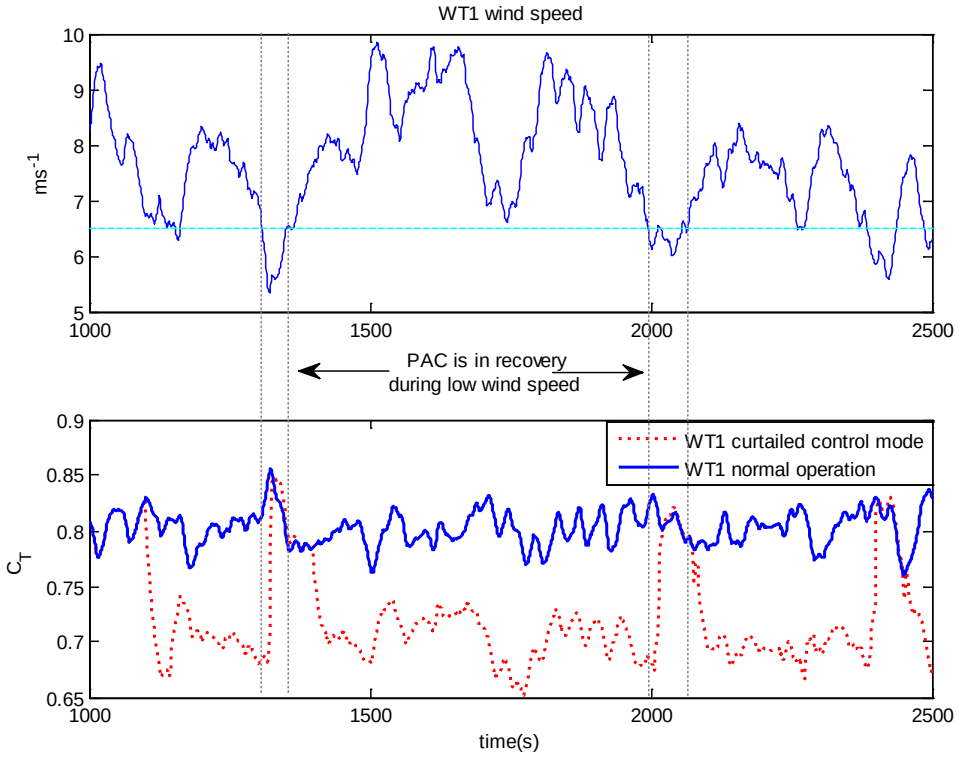

Figure 22: Low wind speed PAC recovery [1].

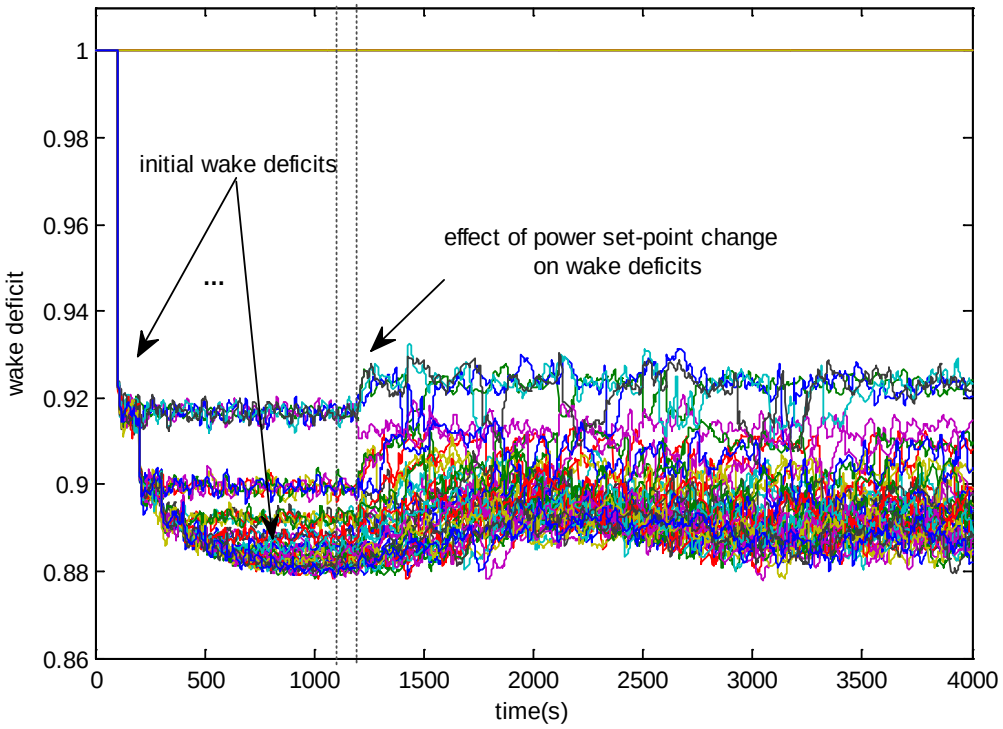

Figure 23: Wake deficits after power curtailment at 1100s [1].

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.

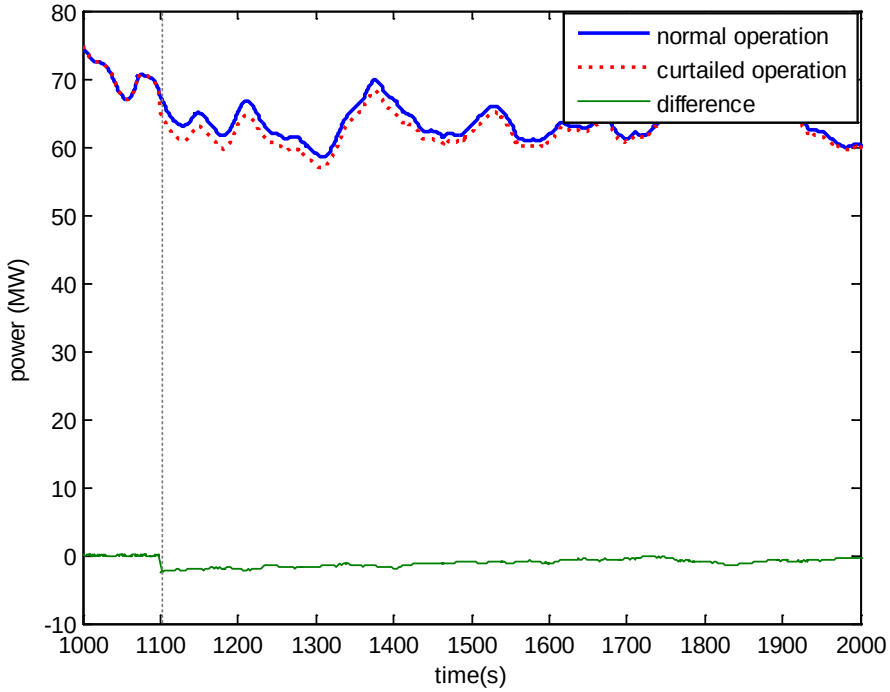

Figure 24: Total power output before and after curtailment of 250MW wind farm in below rated 8 m/s wind speed [1].

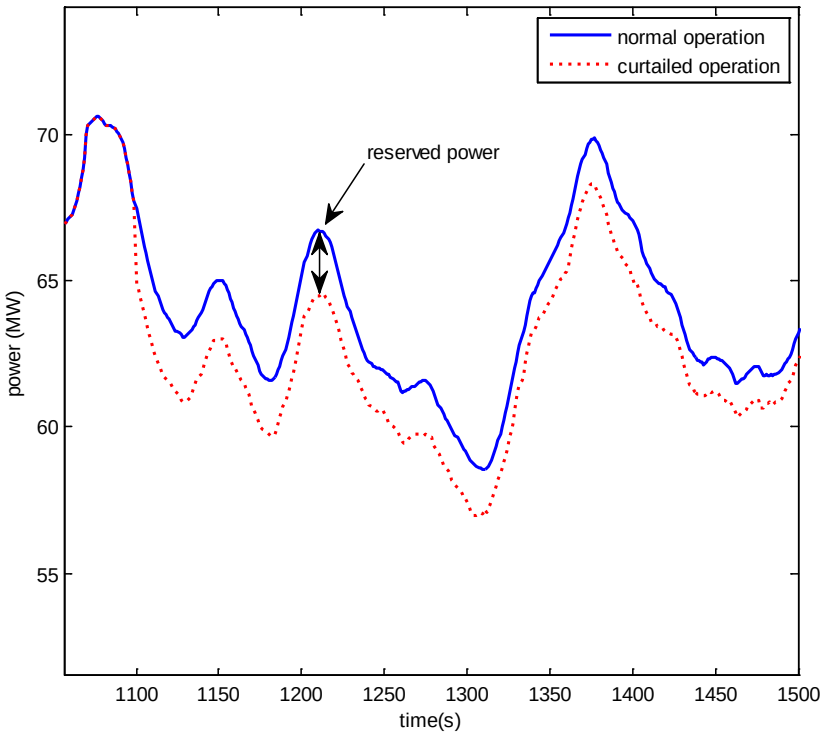

Figure 25: Reserved power at curtailed control mode [1].

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.